

# **SRC Technical Reference Manual**

<b>1. GENERAL .....</b>	<b>1</b>
1.1 FRONT PANEL DIAGRAM .....	1
1.2 BOARD LAYOUT .....	2
1.3 ADDRESS LIST .....	3
1.4 PARTS LIST .....	8
<b>2. VME.....</b>	<b>16</b>
2.1 PIN DESCRIPTION.....	16
2.2 CONSTRAINT FILE.....	17
2.3 PLACEMENT REPORT .....	19
2.4 XACT PERFORMANCE .....	23
2.5 STATE CAD DIAGRAMS.....	25
2.5.1 <i>VME_INT</i> .....	25
2.6 HDL CODE.....	26
2.6.1 <i>VME_INT</i> .....	26
<b>3. MASTER CLOCK .....</b>	<b>34</b>
3.1 PIN DESCRIPTION.....	34
3.2 CONSTRAINT FILE.....	35
3.3 PLACEMENT REPORT .....	39
3.4 XACT PERFORMANCE .....	43
3.5 STATE CAD DIAGRAMS .....	45
3.5.1 <i>VME</i> .....	45
3.5.2 <i>CRD_REV1</i> .....	46
3.5.3 <i>CRW_REV1</i> .....	47
3.5.4 <i>DIV7_REV1</i> .....	48
3.5.5 <i>VBX_REV1</i> .....	49
3.6 HDL CODE.....	50
3.6.1 <i>VME</i> .....	50
3.6.2 <i>CRD_REV1</i> .....	52
3.6.3 <i>CRW_REV1</i> .....	54
3.6.4 <i>DIV7_REV1</i> .....	56
3.6.5 <i>VBX_REV1</i> .....	58

<b>4. RSM.....</b>	<b>60</b>
4.1 PIN DESCRIPTION.....	60
4.2 CONSTRAINT FILE.....	64
4.3 PLACEMENT REPORT .....	67
4.4 XACT PERFORMANCE .....	75
4.5 STATE CAD DIAGRAMS .....	77
4.5.1 <i>VME_INT</i> .....	77
4.5.2 <i>L2A_SM</i> .....	78
4.5.3 <i>RDQ_SM</i> .....	79
4.5.4 <i>READ_Q</i> .....	81
4.5.5 <i>READ_SM6</i> .....	82
4.5.6 <i>RES_PR</i> .....	83
4.5.7 <i>RSMMODES</i> .....	84
4.5.8 <i>SCAN_Q</i> .....	85
4.5.9 <i>SRC_ERR1</i> .....	86
4.5.10 <i>SRC_ERR2</i> .....	87
4.6 HDL CODE.....	88
4.6.1 <i>VME_INT</i> .....	88
4.6.2 <i>L2A_SM</i> .....	91
4.6.3 <i>RDQ_SM</i> .....	94
4.6.4 <i>READ_Q</i> .....	96
4.6.5 <i>READ_SM6</i> .....	98
4.6.6 <i>RES_PR</i> .....	113
4.6.7 <i>RSMMODES</i> .....	115
4.6.8 <i>SCAN_Q</i> .....	121
4.6.9 <i>SRC_ERR1</i> .....	124
4.6.10 <i>SRC_ERR2</i> .....	127
<b>5. BUFFER MANAGER .....</b>	<b>131</b>
5.1 PIN DESCRIPTION.....	131
5.2 CONSTRAINT FILE.....	132
5.3 PLACEMENT REPORT .....	137
5.4 XACT PERFORMANCE .....	139
<b>6. FIB INTERFACE .....</b>	<b>141</b>
6.1 PIN DESCRIPTION.....	141
6.2 CONSTRAINT FILE.....	142
6.3 PLACEMENT REPORT .....	144
6.4 XACT PERFORMANCE .....	150
6.5 STATE CAD DIAGRAMS .....	152
6.5.1 <i>VME</i> .....	152
6.5.2 <i>FIFOS</i> .....	154
6.5.3 <i>GLINK</i> .....	155
6.6 HDL CODE.....	156
6.6.1 <i>VME</i> .....	156
6.6.2 <i>FIFOS</i> .....	158
6.6.3 <i>GLINK</i> .....	168
<b>7. TAXI DECODER .....</b>	<b>171</b>
7.1 PIN DESCRIPTION.....	171
7.2 CONSTRAINT FILE.....	172
7.3 PLACEMENT REPORT .....	174
7.4 XACT PERFORMANCE .....	176
<b>8. TSI EMULATOR .....</b>	<b>178</b>

8.1 PIN DESCRIPTION.....	178
8.2 CONSTRAINT FILE.....	180
8.3 PLACEMENT REPORT .....	182
8.4 XACT PERFORMANCE .....	186
8.5 STATE CAD DIAGRAMS .....	190
8.5.1 <i>VME</i> .....	190
8.5.2 <i>AUTO2</i> .....	191
8.5.3 <i>L2A_ST</i> .....	192
8.5.4 <i>MODES</i> .....	193
8.6 HDL CODE.....	194
8.6.1 <i>VME</i> .....	194
8.6.2 <i>AUTO2</i> .....	196
8.6.3 <i>L2A_ST</i> .....	199
8.6.4 <i>MODES</i> .....	201
<b>9. PIPE CAP .....</b>	<b>204</b>
9.1 PIN DESCRIPTION.....	204
9.2 CONSTRAINT FILE.....	205
9.3 PLACEMENT REPORT .....	206
9.4 XACT PERFORMANCE .....	209
9.5 STATE CAD DIAGRAMS .....	211
9.5.1 <i>PIPE_VME</i> .....	211
9.6 HDL CODE.....	212
9.6.1 <i>PIPE_VME</i> .....	212
<b>10. ERROR LOGGER .....</b>	<b>214</b>
10.1 PIN DESCRIPTION.....	214
10.2 CONSTRAINT FILE.....	215
10.3 PLACEMENT REPORT .....	224
10.4 XACT PERFORMANCE .....	226
10.5 STATE CAD DIAGRAMS.....	228
10.5.1 <i>VME</i> .....	228
10.5.2 <i>RUNMODES</i> .....	229
10.6 HDL CODE.....	230
10.6.1 <i>VME</i> .....	230
10.6.2 <i>RUNMODES</i> .....	233
<b>11. SCHEMATICS .....</b>	<b>236</b>



## **1. General**

### ***1.1 Front Panel Diagram***

---

Title: G:\ssrc.EPS

Creator: AutoCAD LT PSOUT

CreationDate: 1997-07-21

## **1.2 Board Layout**

---

Title: C:\temp\src.EPS  
Creator: AutoCAD LT PSOUT  
CreationDate: 1997-07-22

---

### 1.3 Address List

	July 22, 1997					
Hex Address	Name	Bits	Function	Type	Dir	Comments
<b>VME sub-section 0</b>						
<b>0000</b>	Module code		SRC Module code = \$01	<b>Byte</b>	<b>R</b>	
<b>0002</b>	Revision code		Not used	<b>Byte</b>	<b>R</b>	
<b>0004</b>	FPGA Reset register			<b>Byte</b>	<b>R/W</b>	
		7	Not used			
		6	Reset Error Logger			
		5	Reset Pipe Cap Emulator			
		4	Reset TSI Emulator			
		3	Reset FIB Interface			
		2	Reset Readout State Machine			
		1	Reset Master Clock			
		0	Not used			
<b>Mastclk sub-section 1</b>						
<b>1000</b>	Master Clock Control Register			<b>Byte</b>	<b>R/W</b>	
		[7:5]	Not used			
		4	Enable Laser			Enable pulses to external laser (or scope trigger)
		3	Clear Turn Counter			
		2	Clear Bunch Counter			
		1	Enable Emulator			
		0	Enable Master Clock			
<b>1002</b>	Master Clock Status			<b>Byte</b>	<b>R</b>	
		[7:3]	Not used			
		2	MODE400			400ns Mode detected
		1	MODE132			132ns Mode detected
		0	LOCK			Locked to external CDF clock
<b>1004</b>	Scintillator Mask Register	[2:0]	Scintillator trigger required	<b>Byte</b>	<b>R/W</b>	Disabled when 000
<b>1006</b>	External Trig Delay	[4:0]		<b>Byte</b>	<b>R/W</b>	External Trigger Window delay in RF clocks
<b>1008</b>	External Trig Width	[4:0]		<b>Byte</b>	<b>R/W</b>	External Trigger Window width in RF clocks

<b>1010</b>	XQT	-	Global Execute Generator	-	<b>W</b>	Generates XQT (Data irrelevant)
<b>1100-123C</b>	Master Clock Memory			<b>Byte</b>	<b>R/W</b>	Beam structure emulation
		4	Laser			Pulse external laser or scope trigger
		3	Gap			Tev large abort gap
		2	BZero			Bunch Zero
		1	BXing			Bunch Crossing
		0	Sync			Sync (always=1)
			<b>RSM: subsection 2</b>			
<b>2000</b>	RSM Control Register			<b>Byte</b>	<b>R/W</b>	
<b>2002</b>	RSM Status Register			<b>Byte</b>	<b>R</b>	Scan busy,Read busy,0,0,0,Run,Recover,Halt
<b>2004</b>	VRB Fatal Error Mask			<b>Byte</b>	<b>R/W</b>	
<b>2006</b>	VRB Buffer Count Reg			<b>Byte</b>	<b>R/W</b>	
<b>2008</b>	SVXIII history depth			<b>Byte</b>	<b>R/W</b>	
<b>200A</b>	Digitize counter/timer			<b>Byte</b>	<b>R/W</b>	
<b>2010</b>	Buffer Pointer 0			<b>Byte</b>	<b>R</b>	
<b>2012</b>	Buffer Pointer 1			<b>Byte</b>	<b>R</b>	
<b>2014</b>	Buffer Pointer 2			<b>Byte</b>	<b>R</b>	
<b>2016</b>	Buffer Pointer 3			<b>Byte</b>	<b>R</b>	
<b>2018</b>	EMPTY Fifo			<b>Byte</b>	<b>R</b>	
<b>201A</b>	PENDING Fifo			<b>Byte</b>	<b>R</b>	
<b>2020</b>	Preamp Reset counter/timer			<b>Byte</b>	<b>R/W</b>	
<b>2022</b>	SVXIII Reset counter/timer			<b>Byte</b>	<b>R/W</b>	
<b>2024</b>	Portcard Reset counter/timer			<b>Byte</b>	<b>R/W</b>	
<b>2026</b>	FIB Reset counter/timer			<b>Byte</b>	<b>R/W</b>	
<b>2028</b>	RSM State Number		State number of RSM	<b>Byte</b>	<b>R</b>	Refer to table and state machine
<b>2030</b>	Status Info			<b>Word</b>	<b>R</b>	L2A, L2R, L1PTR, RC
			<b>FIB Int :sub-section 3</b>			
<b>3000</b>	Control Reg			<b>Byte</b>	<b>R/W</b>	
		[7:4]	Mode[3:0]			See mode descriptions
		[3:0]	Sub-Mode[3:0]			
<b>3002</b>	Status Reg			<b>Word</b>	<b>R</b>	
		[15:10]	Not used			
		9	Emulator Fifo1 Full			
		8	Emulator Fifo1 Empty			
		7	Emulator Fifo0 Full			
		6	Emulator Fifo0 Empty			

		5	History Fifo2 Full			
		4	History Fifo2 Empty			
		3	History Fifo1 Full			
		2	History Fifo1 Empty			
		1	History Fifo0 Full			
		0	History Fifo0 Empty			
<b>3004</b>	G-Link Control Reg			<b>Byte</b>	<b>R/W</b>	
		5	Auto			Auto=1/Manual=0 select
		4	G-Link Run / Initialize			SRC to FIB G-Link status (Manual only)
		3	Reset			Reset G-Link (Manual only)
		2	FF			G-Link Fill Frame type (Manual only)
		1	ED			G-Link Enable Data
		0	DAV			G-Link Data Valid (Manual only)
<b>3006</b>	G-Link Status Reg			<b>Byte</b>	<b>R</b>	
		[7:5]	Not used			
		4	SRC/FIB link locked			SRC/FIB closed loop lock
		3	G-Link Tx only locked			Local Tx Transmitter lock
		2	STAT1(Non Fatal Error)			INIT/RUN dual status line from FIB
		1	STAT0(Fatal Error)			INIT/RUN dual status line from FIB
		0	POK			FIB Power OK (Status from FIB)
<b>3008</b>	FIB History Fifo (lower word)			<b>Word</b>	<b>R/W</b>	
<b>300A</b>	FIB History Fifo (upper byte)			<b>Byte</b>	<b>R/W</b>	
<b>300C</b>	Fib Emulator Fifo			<b>Word</b>	<b>R/W</b>	
<b>300E</b>	Not used					
<b>3010</b>	Not used					
<b>3012</b>	Not used					
<b>3014</b>	Not used					
<b><u>TSI Emulator : sub-section 4</u></b>						
<b>4000</b>	TSI_Em Control Reg			<b>Byte</b>	<b>R/W</b>	
		[7:4]	Mode			OFF, MANUAL,FIFO_EM, etc
		[3:0]	Sub-Mode			Details
<b>4002</b>	Status Reg			<b>Byte</b>	<b>R</b>	
		7				
		6				
		5				
		4				
		3	ERROR			Global status line to TSI
		2	WAIT			Global status line to TSI
		1	L1_DONE			Global status line to TSI
		0	DONE			Global status line to TSI
<b>4004</b>				<b>Word</b>	<b>R/W</b>	

<b>4006</b>	Taxi Emulator Fifo			<b>Word</b>	<b>R/W</b>	
<b>4008</b>	L1 Rate			<b>Word</b>	<b>R/W</b>	
<b>400A</b>	L1 Trigger Delay			<b>Byte</b>	<b>R/W</b>	
<b>400C</b>	L2 Rate			<b>Word</b>	<b>R/W</b>	
<b>400E</b>	L2A Rate			<b>Byte</b>	<b>R/W</b>	Number of L2R's before a L2A
<b>4010</b>	TSI Emulator Data			<b>Word</b>	<b>R</b>	
<b>4012</b>	TSI Emulator Manual Low word			<b>Word</b>	<b>R/W</b>	
	[15:14]	L1A_ADD[1:0]				Level 1 Accept pointer
	13	L1A				Level 1 Accept
	[12:9]	EID[3:0]				4-bit Event ID
	[8:6]	CALIB[2:0]				Calibration code
	5	CAL				Calibration command
	4	CONTROL				Control command
	3	HALT				Global HALT command
	2	RECOVER				Global RECOVER command
	1	RUN				Global RUN command
	0	TEST				TEST command
<b>4014</b>	TSI Emulator Manual High byte			<b>Byte</b>	<b>R/W</b>	
	[8:6]	Not used				
	[5:4]	L2R_ADD[1:0]				Level 2 Reject pointer
	3	L2R				Level 2 Reject
	[2:1]	L2A_ADD[1:0]				Level 2 Accept pointer
	0	L2A				Level 2 Accept
<b>Pipe Cap Emulator Sub-Section</b>						
<b>5000</b>	Pipe Cap Control Register					
	<b>Error Logger Sub-section</b>					
<b>6000</b>	Control Register			<b>Byte</b>	<b>R/W</b>	
		[7:4]	Mode			
		[3:0]	Sub-Mode			
<b>6002</b>	Status Register			<b>Word</b>	<b>R</b>	
		9	Fifo4 Almost Full			Error fifo flags
		8	Fifo4 Empty			
		7	Fifo3 Almost Full			
		6	Fifo3 Empty			
		5	Fifo2 Almost Full			
		4	Fifo2 Empty			
		3	Fifo1 Almost Full			
		2	Fifo1 Empty			
		1	Fifo0 Almost Full			
		0	Fifo0 Empty			

<b>6004</b>	Fifo Error Mask register			<b>Word</b>	<b>R/W</b>	Error is recorded on mask satisfied
<b>6006</b>	Error Fifo 1,0			<b>Word</b>	<b>R</b>	
<b>6008</b>						
<b>600A</b>	Error Fifo 2	[7:0]	BCN[7:0]	<b>Byte</b>	<b>R</b>	Bunch Counter time stamp
<b>600C</b>	Error Fifo 4,3	[15:0]	TURN[15:0]	<b>Word</b>	<b>R</b>	Turn counter time stamp
<b>6010</b>	Error Counter 0	[13:0]	SRC_ERR_0_[13:0]	<b>Word</b>	<b>R</b>	
<b>6012</b>	Error Counter 1	[13:0]	SRC_ERR_1_[13:0]	<b>Word</b>	<b>R</b>	
<b>6014</b>	Error Counter 2	[13:0]	SRC_ERR_2_[13:0]	<b>Word</b>	<b>R</b>	
<b>6016</b>	Error Counter 3	[13:0]	SRC_ERR_3_[13:0]	<b>Word</b>	<b>R</b>	
<b>6018</b>	Error Counter 4	[13:0]	SRC_ERR_4_[13:0]	<b>Word</b>	<b>R</b>	
<b>601A</b>	Error Counter 5	[13:0]	SRC_ERR_5_[13:0]	<b>Word</b>	<b>R</b>	
<b>601C</b>	Error Counter 6	[13:0]	SRC_ERR_6_[13:0]	<b>Word</b>	<b>R</b>	
<b>601E</b>	Error Counter 7	[13:0]	SRC_ERR_7_[13:0]	<b>Word</b>	<b>R</b>	
<b>6020</b>	Error Counter 8	[13:0]	VRB_ERR_0_[13:0]	<b>Word</b>	<b>R</b>	
<b>6022</b>	Error Counter 9	[13:0]	VRB_ERR_1_[13:0]	<b>Word</b>	<b>R</b>	
<b>6024</b>	Error Counter 10	[13:0]	VRB_ERR_2_[13:0]	<b>Word</b>	<b>R</b>	
<b>6026</b>	Error Counter 11	[13:0]	VRB_ERR_3_[13:0]	<b>Word</b>	<b>R</b>	
<b>6028</b>	Error Counter 12	[13:0]	VRB_ERR_4_[13:0]	<b>Word</b>	<b>R</b>	
<b>602A</b>	Error Counter 13	[13:0]	VRB_ERR_5_[13:0]	<b>Word</b>	<b>R</b>	
<b>602C</b>	Error Counter 14	[13:0]	VRB_ERR_6_[13:0]	<b>Word</b>	<b>R</b>	
<b>602E</b>	Error Counter 15	[13:0]	VRB_ERR_7_[13:0]	<b>Word</b>	<b>R</b>	

#### 1.4 Parts List

	CDF-SRC pcboard			
	6/3/97			
	part #	QTY	package	comments
C1	2uF tantalum	1		
C12	0.1uF	1		
C13	1000pF	1		
C14	1000pF	1		
C15	0.1uF	1		
C16	0.1uF	1		
C2	6.8uF tantalum	1		
C3	6.8uF tantalum	1		
C4	10uF tantalum	1		
C5	10uF tantalum	1		
C6	0.1uF	1		
C7	0.1uF	1		
C8	short	1		
CAP1	0.1uF bypass	1		
CAP11	0.1uF	1		
CAP19	0.47uF	1		Vtt bypass caps
CAP20	0.47uF	1		"
CAP4				series caps for clocks
CAP5				
D1	1N914	1		1N914
D2	1N914	1		
D3	1N914	1		
D4	1N914	1		
D5	1N914	1		
D6	1N914	1		
DC	0.1uF bypass caps	11		
DC	0.1uF bypass	22		
DC	0.1uF bypass	17		
DC	bypass	4		at least
DC124-133	0.1uF bypass caps	10	Vee	
DC92-123	0.1uF bypass caps	43	Vcc	
Del1	DDU222-F	1	SIP8	250ns total delay

Del2	DDU224-F	1	SIP14	25ns total delay
Del3	DDU224-F	1		25ns total delay
Del5	DDU-224F	1		100ns total delay
J10	single row 9 pins	1		
J100	AMP 3-520459-3	1		mastclk
J101	AMP 3-520459-3	1		
J102	fr panel lemo	1		
J103	fr panel lemo	1		
J104	fr panel lemo	1		
J12	single row 9 pins	1		
J13	single row 9 pins	1		
J14	fr panel lemo	1		
J200	AMP 1-520459-3	1		rsm
J202	50 pin straight berg	1		from RSM page
J3	single row 9 pins	1		
J300	AMP 5-520459-3	1		fib_int
J4	single row 9 pins	1		
J5	fr panel lemo	1		
J6	single row 9 pins	1		
J7	single row 9 pins	1		
J8	single row 9 pins	1		
J9	single row 9 pins	1		
Jmpr1	single row 3 pins	1		
Jmpr10	single row 3 pins	1		
Jmpr11	single row 3 pins	1		
Jmpr12	single row 3 pins	1		
Jmpr13	dual berg	1		clk invert jumper
Jmpr14	dual berg 20	1		
Jmpr15	dual berg	1		
Jmpr2	dual berg 14	1		
Jmpr21	dual berg	1		
Jmpr22	dual berg	1		
Jmpr23	dual berg	1		
Jmpr24	dual berg	1		
Jmpr25	dual berg	1		
Jmpr26	dual berg	1		
Jmpr27	dual berg	1		
Jmpr28	dual berg	1		
Jmpr3	single row 3 pins	1		
Jmpr4	dual berg 20	1		
Jmpr5	single row 3 pins	1		
Jmpr7	3 pin berg	1		

Jmpr8	single row 3 pins	1		
Jmpr9	single row 3 pins	1		
LED1	dialight 555-3009	1		dialight 555-3007 or 3009
LED10	dialight 564-0300-132	1		"
LED11	dialight 564-0300-132	1		"
LED12	dialight 564-0300-132	1		"
LED13	dialight 564-0300-132	1		"
LED14	dialight 564-0300-132	1		lock/mode LED
LED15	Dialight 550-3507	1		Vee LED
LED16	Dialight 550-3507	1		Vcc LED
LED17	Dialight 550-3507	1		ack/err LED
LED18	Dialight 550-3507	1		mode LED
LED19	Dialight 550-3507	1		fatal error LED
LED2	dialight 555-3009	1		
LED20	dialight 555-3009	1		
LED21	dialight 555-3009	1		
LED22	Dialight 550-3507	1		Vtt LED
LED23	dialight 555-3009	1	dip8	
LED24	Dialight 550-3507	1		global address LED
LED25	dialight 555-3009	1		
LED26	Dialight 550-3507	1		gl_locked LED
LED3	dialight 555-3009	1		
LED4	dialight 555-3009	1		
LED5	dialight 555-3009	1		
LED6	dialight 564-0300-132	1		Queue indicators
LED7	dialight 564-0300-132	1		L1
LED8	dialight 564-0300-132	1		L2
LED9	dialight 564-0300-132	1		L3
P0	352009-1	1	Fermi supply	AMP 5-row, 95-pin, 2mm, AMP
P1	0201-160-2101	1		5-row,160-pin DIN bkpln conn,Harting
P2	0201-160-2101	1		
R1	200ohm 1/4W, 5%			
R10	20KSip6Res5	1	SIP6	
R101	200ohm 1/4W, 5%	1		terminators for rsm_clkC
R102	200ohm 1/4W, 5%	1		
R103	200ohm 1/4W, 5%	1		terminators for rf_out B
R104	200ohm 1/4W, 5%	1		
R105	200ohm 1/4W, 5%	1		
R11	10K 1/4W, 5%	1	SIP6	
R12	20K Sip6Res5	1	SIP6	
R13	10K 1/4W, 5%	1		

R14	200ohm 1/4W, 5%			
R15	100ohm 1/4W, 5%			glink clock terminator options
R16	omit			
R17	omit			
R18	omit			
R19	omit			
R2	51K 5%, 1/4W	1		
R20	10K 1/4W, 5%	1		
R21	20KSip6Res5	1		
R22	1K Sip10Res9	1	SIP10	
R23	1K Sip10Res9	1		
R24	51ohm 1/4W, 5%			
R25	100ohm Sip8Res4	1	SIP8	
R26	200ohm 1/4W, 5%	1		terminators for rsm_clkA
R27	1K Sip10Res9	1		
R28	1K Sip10Res9	1		
R29	220ohm 1/4W, 5%	1		TBD parts around receiver
R3	20KSip10Res9	1	SIP10	
R30	100ohm 1/4W, 5%	1		chip
R31	130ohm 1/4W, 5%	1		
R32	360ohm 1/4W, 5%	1		
R33	220ohm 1/4W, 5%	1		
R34	10K 1/4W, 5%	1		for prom dnld
R35	20K Sip6Res5	1		
R36	10K 1/4W, 5%	1		
R37	10K 1/4W, 5%	1		
R38	10K 1/4W, 5%	1		
R39	20K Sip6Res5	1		
R4	20KSip10Res9	1		
R40	200ohm 1/4W, 5%	1		terminators for rf_outA
R41	10K 1/4W, 5%	1		buffer enable pullups
R42	1K Sip10Res9	1	SIP10	
R43	1K Sip10Res9	1	SIP10	
R44	10K 1/4W, 5%	1		/reset_errlog pull up
R45	20K Sip6Res5	1	SIP6	
R46	390ohm 1/4W, 5%	1		
R47	390ohm 1/4W, 5%	1		
R48	390ohm 1/4W, 5%	1		
R49	4.7K 1/4W, 5%	1		
R5	200ohm 1/4W, 5%			
R50	4.7K 1/4W, 5%	1		
R51	4.7K 1/4W, 5%	1		

R52	10K 1/4W, 5%	1		
R53	20K Sip6Res95	1		
R54	omit			
R55	200ohm 1/4W, 5%	1		
R56	100ohm 1/4W, 5%	1		
R57	100K 1/4W, 5%	1		
R58	100K 1/4W, 5%	1		
R59	66.5K 1/4W, 1%	1		
R6	470ohm Sip6Res5	1		
R60	66.5K 1/4W, 1%	1		
R61	66.5K 1/4W, 1%	1		
R62	66.5K 1/4W, 1%	1		
R63	2K 1/4W, 5%	1		
R64	2K 1/4W, 5%	1		
R65	2K 1/4W, 5%	1		
R66	200ohm 1/4W, 5%	1		
R67	10K 1/4W, 5%	1		
R68	20K Sip6Res5	1	SIP6	
R69	10K 1/4W, 5%	1		
R7	200ohm 1/4W, 5%			
R70	10K 1/4W, 5%	1		/reset_bm pull up
R71	300ohm 1/4W,5%			
R72	10K 1/4W, 5%	1		
R73	10K 1/4W, 5%	1		
R74	51ohm 1/4W, 5%	1		
R75	51ohm 1/4W, 5%	1		
R76	51ohm 1/4W, 5%	1		
R77	10K 1/4W, 5%	1		
R78	10K 1/4W, 5%	1		
R79	300ohm Dip16Res8	1	DIP16	various values
R8	100ohm Sip8Res4	1		
R80	10K 1/4W, 5%	1		
R81	20K Sip6Res5	1		
R82	300ohm Dip16Res8	1		
R83	300ohm Dip16Res8	1	"	
R84	10K 1/4W, 5%	1		
R85	300ohm 1/4W,5%			
R89	470ohm 1/4W, 5%			
R9	10K 1/4W, 5%			
R90	200ohm 1/4W, 5%			
R91	51ohm 1/4W, 5%	1		
R92	51ohm 1/4W, 5%	1		

R93	100ohm Dip16Res8	1		
R94	200ohm 1/4W, 5%	1		terminators for rsm_clkA
R95	200ohm 1/4W, 5%	1		terminators for rf_outA
R96	10K 1/4W, 5%	1		"
R97	10K 1/4W, 5%	1		
R98	200ohm 1/4W, 5%			
R99	200ohm 1/4W, 5%			
REG1	LM337	1		negative voltage regulator for Vtt
RN1	100ohm Dip16Res8	1	DIP16	
RN2	100ohm Dip16Res8	1	DIP16	
RN3	470ohm Sip10Res9	1		
RN4	470ohm Sip10Res9	1		
RN5	470ohm Sip10Res9	1		
RN6	1K Sip6Res5			
RN7	1K Sip6Res5			
RN8	10K Dip16Res8	1		
RN9	10K Dip16Res8	1		
RP1	10K 22 turn	1		
RP2	10K 22 turn	1		
TX1	60TXGLNK	1	60-PIN	special conn/supplied by Fermilab???
U1	74ACT520	1	DIP20	8 bit comparator
U10	74ACT520	1		8 bit comparator
U11	74ACT520	1		
U12	74ACT520	1		
U13	74ACT520	1		
U14	74ACT245	1	DIP20	octal bus tranceiver
U15	74ACT245	1		
U16	74ACT04	1	DIP14	
U17	XC3130APC84-3	1	PLCC84	VME
U18	XC1765D	1	DIP8	for XC3130A
U19	74ACT244	1	DIP20	octal buffer
U2	74ACT00	1	DIP14	
U20	74ACT244	1		
U21	74ACT244	1		
U22	CY7C470	1	PLCC32	8K x 9 fifo
U23	CY7C470	1		
U24	CY7C470	1		
U25	CY7C470	1		
U26	CY7C470	1		
U27	SY100S324JC	1	PLCC28	ttl/ecl translator-Synergy mfg
U28	MC10H601	1	PLCC28	ecl/ttl translator-Motorola

U29	DS90C031	1	SOIC16	driver
U3	74LS14	1		Hex Schmitt Trigger
U30	DS90C032	1	SOIC16	receiver
U31	MC10H604	1	PLCC28	ttl/ecl translator
U32	MC10H604	1		
U33	MC10H604	1		
U34	MC10H604	1		
U34	MC10H604	1		
U36	CY7C470	1	PLCC32	
U37	CY7C470	1		
U38	CY7C470	1		
U39	CY7C470	1		
U4	74ACT374	1	DIP20	D-type FF or latches
U40	CY7C470	1		
U41	74ACT244	1	DIP20	
U42	74ACT244	1		
U43	74ACT244	1		
U44	74ACT244	1		
U45	74ACT244	1		
U46	XC3164APG132-2	1	PGA132	Fib_Int
U47	XC1765D	1	DIP8	prom for XC3164A
U48	XC3195APG175-3	1	PGA175	Err_Log
U49	XC17128D	1		prom for XC3195A
U5	74ACT374	1		
U50	XC3195APG175-3	1	PGA175	Taxi
U51	DS90C032	1	SOIC16	receiver
U52	DS90C032	1		
U53	DS90C032	1		
U54	DS90C031	1	SOIC16	driver
U55	DS90C031	1		
U56	DS90C031	1		
U57	DS90C031	1		
U58	CY7C421	1	PLCC32	512 x 9 fifo
U59	CY7C421	1		
U6	74ACT374	1		
U60	CY7C421	1		
U61	74ACT373	1	DIP20	octal latch
U62	DS90C031	1	"	
U63	XC3164APC84-2	1	PLCC84	Buf_Mng
U64	CY7C470	1	PLCC32	
U65	XC4013EPG223-4	1	PGA223	RSM
U66	74ACT240	1		

U67	269050-1 (fiber optic)	1	DIP14	Amp receiver
U68	Am7969	1	PLCC28	receiver
U69	74ACT04	1	DIP14	
U7	74ACT244	1	DIP20	octal buffer
U70	XC17128D	1	DIP8	prom for XC3195A
U71	74ACT221	1	DIP14	one shot
U72	XC1765D	1	DIP8	prom for XC3164A
U73	MC10H604	1		
U74	74ACT244	1	DIP20	
U75	DS90C032	1	SOIC16	
U76	DS90C032	1	SOIC16	
U77	AD9696	1	DIP8	comparator
U78	AD9696	1		
U79	AD9696	1		
U8	74S38	1	DIP14	nand gate
U80	XC1765D	1	DIP8	prom for XC3164A
U81	AD790	1	DIP8	comparator
U82	74ACT244	1		
U83	XC4010EPC84-3	1	PLCC84	Pipe_cap
U84	XC3164APG132-2	1	PGA132	Mastclk
U85	AD820	1	DIP8	FET input opamp
U86	CY7C199	1	DIP28	32K x 8 static RAM
U87	XC17256D	1	DIP8	prom for 4010E PC84
U88	74ACT244	1		
U89	74ACT244	1	DIP20	
U9	74ACT04	1	DIP14	
U90	74ACT244	1		
U91	74ACT244	1		
U93	17256D	1	DIP8	prom for XC4013E PG223
U94	XC3130APC84-3	1	PLCC84	Taxi_dec
U95	XC1765D	1	DIP8	prom for XC3130A
U96	74ACT244	1	DIP20	
X1	53MHz	1	DIP14	crystal

## 2. VME

### 2.1 Pin Description

Design				SRC/VME	
Date	I	O	OT	I/O	Function
					<b>18-Jul-97</b>
					3:39 PM
<b>Signal</b>	<b>I</b>	<b>O</b>	<b>OT</b>	<b>I/O</b>	
/MCM		1			Master clock memory request
/MCLK		1			Master clock request
/RSM		1			RSM request
/FIB_INT		1			FIB Interface request
/TAXI		1			Taxi Interface request
/PIPE_CAP		1			Pipe Cap Emulator request
/ERR_LOG		1			Error Logger request
/VMEOK[6:1]	6				Local acknowledge
/RES_RSM		1			Reset RSM FPGA
/RES_FIBINT		1			Reset FIB_INT FPGA
/RES_TAXI		1			Reset TAXI (TSI) FPGA
/RES_PIPE		1			Reset PIPE_CAP FPGA
/RES_ERRLOG		1			Reset ERR_LOG FPGA
BDS[1:0]	2				Data strobe inputs
/STROBE[1:0]		2			Data strobe outputs
/BWRITE	1				VME Write
/CONFIG0		1			Configuration indicator
/ACK		1			VME Acknowledge
/ERR		1			VME Bus Error
/BIACK	1				VME Interrupt Acknowledge
/BLWORD	1				VME Longword request
BAM[5:0]	6				Address modifier code
VADDR[14:1]	14				Address lines
VDAT[7:0]				8	VME data lines (lower byte only)
LADDR	1				Latch address
/ENDAT		1			Enable datac
/TML	1				That's me lower
/TMM	1				That's me middle
/TMG	1				That's me global
/TMU	1				That's me upper
GL_ADDR	1				Geographical address
/GA_IND		1			Geographical address indicator
DEL_OUT		1			Delay line output
DEL3,DEL4,DEL5	3				Delay line inputs
	40	20	0	8	68

## 2.2 Constraint File

```
# Design: vme
# Created by XACT Floorplanner ver 6.0.0
NOTPLACE BLOCK *: PAD16 PAD17 PAD22 PAD60 PAD51;
PLACE BLOCK LADDR: P13;
PLACE BLOCK GL_ADDR: P14;
PLACE BLOCK DEL_OUT: P80;
PLACE BLOCK -PIPE_CAP: P71;
PLACE BLOCK -GA_IND: P52;
PLACE BLOCK -FIB_INT: P70;
PLACE BLOCK -ERR_LOG: P69;
PLACE BLOCK -CONFIG0: P53;
PLACE BLOCK WRITE: [BF DJ];
PLACE BLOCK TM: [BF DJ];
PLACE BLOCK OK: [BF DJ];
PLACE BLOCK LWORD: [BF DJ];
PLACE BLOCK IACK: [BF DJ];
PLACE BLOCK ERR: [BF DJ];
PLACE BLOCK ACK: [BF DJ];
PLACE BLOCK $I36/I35: [BF DJ];
PLACE BLOCK $IN421: [BF DJ];
PLACE BLOCK VDAT7: P23;
PLACE BLOCK VDAT6: P24;
PLACE BLOCK VDAT5: P25;
PLACE BLOCK VDAT4: P26;
PLACE BLOCK VDAT3: P27;
PLACE BLOCK VDAT2: P28;
PLACE BLOCK VDAT1: P29;
PLACE BLOCK VDAT0: P30;
PLACE BLOCK VADDR14: P4;
PLACE BLOCK VADDR13: P5;
PLACE BLOCK VADDR12: P6;
PLACE BLOCK VADDR11: P7;
PLACE BLOCK VADDR10: P8;
PLACE BLOCK VADDR9: P9;
PLACE BLOCK VADDR8: P10;
PLACE BLOCK VADDR7: P11;
PLACE BLOCK VADDR5: P15;
PLACE BLOCK VADDR4: P16;
PLACE BLOCK VADDR3: P17;
PLACE BLOCK VADDR2: P18;
PLACE BLOCK VADDR1: P19;
PLACE BLOCK DEL5: P75;
PLACE BLOCK DEL4: P76;
PLACE BLOCK DEL3: P77;
PLACE BLOCK BS1: P83;
PLACE BLOCK BS0: P84;
PLACE BLOCK BAM5: P34;
PLACE BLOCK BAM4: P35;
PLACE BLOCK BAM3: P36;
PLACE BLOCK BAM2: P37;
PLACE BLOCK BAM1: P38;
PLACE BLOCK BAM0: P39;
```

```
PLACE BLOCK -VMEOK6: P46;
PLACE BLOCK -VMEOK5: P47;
PLACE BLOCK -VMEOK4: P48;
PLACE BLOCK -VMEOK3: P49;
PLACE BLOCK -VMEOK2: P50;
PLACE BLOCK -VMEOK1: P51;
PLACE BLOCK -TMU: P40;
PLACE BLOCK -TMM: P41;
PLACE BLOCK -TML: P44;
PLACE BLOCK -TMG: P45;
PLACE BLOCK -TAXI: P73;
PLACE BLOCK -RSM: P68;
PLACE BLOCK -MCM: P67;
PLACE BLOCK -MCLK: P66;
PLACE BLOCK -LWORD: P58;
PLACE BLOCK -ERR: P59;
PLACE BLOCK -ENDAT: P63;
PLACE BLOCK -BWRITE: P62;
PLACE BLOCK -BIACK: P56;
PLACE BLOCK -ACK: P57;
PLACE BLOCK -STROBE1: P3;
PLACE BLOCK -STROBE0: P2;
# End
```

### **2.3 Placement Report**

PLACEMENT RESULTS FOR DESIGN VME  
From PPR Version 5.2.1

1997/03/17 09:37:23

Xilinx, Inc.  
(c) Copyright 1997. All Rights Reserved.

#### Report Contents

---

1. Device Utilization
2. Guided Placement Summary
3. Unguided Blocks from Input Design
4. Implementation Options

#### Device Utilization

---

Partitioned Design Utilization Using Part 3130APC84-3

---

	No.	Used	Max Available	%Used
Occupied CLBs	41	100	41%	
Bonded I/O Pins	69	74	93%	
CLB Function Generators (*)	47	200	23%	
CLB Flip Flops	11	200	5%	
IOB Input Flip Flops	4	80	5%	
IOB Output Flip Flops	0	80	0%	
3-State Buffers	0	220	0%	
3-State Longlines	0	20	0%	

(\*) Each base F or FGM function counts as two

#### CPU Times

CPU time taken for Placement: 0 hrs 0 mins 31 secs

#### Guided Placement Summary

The following table summarizes the guided placement process. For each type of LCA block, this table shows (1) how many blocks of that type exist in the input design file; (2) how many of these blocks were matched to blocks in the guide file; and (3) how many of the matched blocks were used to guide the placement of the input design.

A matched block will not be used to guide placement if any one of the following is true:

- \* The guide\_blk option is set to ROUTED\_ONLY and the matched block does not have any routing connected to it in the guide LCA file.
- \* The matched block is placed elsewhere by a constraint in the input design or the CST file.
- \* A different block is placed in the matched block's guide location by a constraint in the input design or the CST file.
- \* The matched block is part of an RLOC set and other blocks in that set were not matched. A partially-matched RLOC set will be guided only if there is space available for the unmatched blocks AND if the structure of the RLOC set can be respected.
- \* A pin on a matched block does not have any routing connected to it in the guide LCA file, so the pin was swapped with another pin on that CLB to improve the routing of the associated signal(s). For example, the X and Y flip-flops may be swapped if their output pins are not routed.

Block Type	In Design	Matched in Guide	Placement Guided
CLB flip-flop	11	11	10
CLB base FG function	39	21	17

CLB base F function	8	2	2
CLB base FGM function	0	0	0
3-state buffer	0	0	0
I/O pad	69	69	69

If there are blocks in the input design which could not be matched in the guide file they will be listed in the "Unguided Blocks" chapter below.

### Unguided Blocks from Input Design

---

The following lists show which blocks in the input design were not matched to blocks in the guide file and therefore were NOT guided during the placement process.

CLB flip flops driving following Q output signals:

---

ACK

CLB function generators driving following output signals:

---

Base Output Signal Name

---

```
F CLKACK
F $1I427/_111_
F $1I427/_113_
F $1I427/_114_
F $1I427/_121_
F $1I427/_126_
FG IACK
FG WRITE
FG MCLC
FG $1I427/_124_
FG MCM
FG $1I427/_56_
FG $1I427/_116_
FG $1I427/_138_
FG CODE
FG MASK
FG DEL
FG ENDAT
FG CLKERR
FG $1N714
FG $1N761
FG STROBE0
FG STROBE1
FG $1I427/_57_
FG $1I427/_58_
FG $1I427/_112_
FG $1I427/_128_
FG $1I427/_137_
```

Implementation Options

---

### PPR Parameters

```
Design      = vme
Parttype    = from design file
LogFile     = ppr.log
Outfile     = <design name>
Estimate    = FALSE
```

### Additional Specified or Non-Default Parameters

```
paramfile   = params.txt
cstfile    = D:\CDF\SRC\VME\xproject\v1_0\rev1\vme.cst
seed        = 858591402
design      = vme
placer_effort = 3
router_effort = 3
path_timing   = true
guide        = d:\cdf\src\vme\xproject\v1_4\rev1\vme.lca
route_thru_bufg = ok
route_thru_blk = ok
guide_blk = all
lock_routing  = whole_sigs
split_report  = true
```

### Parameter Values from XACTINIT.DAT

```
ORCAD_NAMES      = false
```

---

===== End of Report =====

## 2.4 XACT Performance

### XACT PERFORMANCE RESULTS FOR DESIGN VME From PPR Version 5.2.1

1997/03/17 09:42:10

Xilinx, Inc.

(c) Copyright 1997. All Rights Reserved.

Xact Performance Summary

---

Parttype Used : 3130APC84

Speed Grade : -3

```

End-
Limit Actual Points
(ns) * (ns) Missed Specification
-----
<auto> 46.3 0/18 DEFAULT_FROM_FFS_TO_FFS=FROM:ffs:TO:ffs
<auto> 47.8 0/39 DEFAULT_FROM_PADS_TO_FFS=FROM:pads:TO:ffs
<auto> 53.6 0/27 DEFAULT_FROM_FFS_TO_PADS=FROM:ffs:TO:pads

```

(\*) Use the -FailedSpec and/or -TSMMaxPaths options of the XDelay-TimeSpec command, accessible through the XDE or XDelay program, to confirm the actual path delays computed by PPR. Note that XDelay-TimeSpec will not report paths that start and end in the same block (CLB or IOB) and use no external routing.

Note: The design contains more than one clock signal (\$1N560, DELAY3, STROBE0, CLKERR, CLKACK). Default specs generated automatically by PPR will not control each clock at its own best possible frequency; clocks which could run faster may be controlled at a slower frequency. If you have specified no TIMESPEC requirements (or only a generic 'FROM:FFS:TO:FFS' requirement), the design performance may not have been optimized by PPR. For best results, supply individual TIMESPECS for the flip-flops on each clock signal that you care about.

\*\*\* PPR: WARNING 7028:

The design has flip-flops with asynchronous set/reset controls (PRE/SD or CLR/RD pins). When PPR analyzes design timing, it does not trace paths through the asynchronous set/reset input and on through the Q output.

If you want PPR to control the delay on paths through asynchronous set/reset pins, you must split the delay requirement into two segments: one ending at the set/reset input, and the other beginning at the flip-flop output. If you want PPR not to analyze paths that lead to asynchronous set/reset pins, attach an IGNORE specification to the pin(s) or signal(s).

By default, XDelay reports all paths through asynchronous set/reset pins. To prevent XDelay from showing these paths, use FlagBlk CLB\_Disable\_SR\_Q on the appropriate flip-flops.

===== End of Report =====

## 2.5 State CAD Diagrams

### VME\_INT

\CD\FSRC\VME\vme\_int.dia  
SRC VME Interface  
31 Oct. 1996  
Updated March 9, 1997

```

ADDR[] = 14:0

CODE = ADDR[0] & !WRITE & TM
MASK = ADDR[4] & TM
PAD = !WRITE & TM & (ADDR[0] # ADDR[4])

MCLK = (ADDR>=^h1000)&(ADDR<^h1100) & TM
MCM = (ADDR>=^h1100)&(ADDR<=^h123E) & TM
RSM = (ADDR>=^h2000)&(ADDR<^h3000) & TM
FIB_INT = (ADDR>=^h3000)&(ADDR<^h4000 )& TM
TAXI = (ADDR>=^h4000)&(ADDR<^h5000) & TM
PIPE_CAP = (ADDR>=^h5000)&(ADDR<^h6000) & TM
ERR_LOG = (ADDR>=^h6000)&(ADDR<^h7000) & TM

Check address modifier, Use Fermilab requested
09, 0A, 0B, and HUHEPL favorites 29, 2D, 39 & 3D.

AM[] = 5:0
%AMOK% = ( (AM=^h09)#(AM=^h0A)#(AM=^h0B)
# (AM=^h29) # (AM=^h2D) #(AM=^h39) # (AM=^h3D) )

%D0_IT% = ( TM & !ACK & !LWORD )

%ADDR_VALID% = ( (CODE # MASK # MCLK # MCM # RSM # FIB_INT# TAXI #
PIPE_CAP # ERR_LOG) & (!LWORD) & (!ACK) )

If board is decoded then enable the VME data bus
and start the delay line on either STROBE

DEL = (DO_IT) & (S0 # S1) & (AMOK)
ENDAT = (DO_IT) & (LADDR) & (AMOK)

If board is decoded, bang the appropriate STROBE
lines

STROBE0 = (ADDR_VALID) & (AMOK) &
((LADDR & !WRITE) # (DEL4 & !DEL5 & WRITE))

STROBE1 = (ADDR_VALID) & (AMOK) & S1 &
((LADDR & !WRITE) # (DEL4 & !DEL5 & WRITE))

If the address has responded by DEL5, then ACKNOWLEDGE,
otherwise hit the electric dog collar button!

CLK_ACK = (OK#CODE#MASK) & DEL5
& (ADDR_VALID)

CLK_ERR = DEL5 & !(OK#CODE#MASK)

Bye now!

```

\CDF\SRCS\VME\vmem\_int.dia  
SRC VME Interface  
31 Oct. 1996  
Updated March 9, 1997

```
ADDR[] = 14:0

CODE = ADDR[0] & !WRITE & TM
MASK = ADDR[4] & TM
MCLK = (ADDR>=^h1000)&(ADDR<^h1100) & TM
MCM = (ADDR>=^h1100)&(ADDR<=^h123E) & TM
RSM = (ADDR>=^h2000)&(ADDR<^h3000) & TM
FIB_INT = (ADDR>=^h3000)&(ADDR<^h4000 )& TM
TAXI = (ADDR>=^h4000)&(ADDR<^h5000) & TM
PIPE_CAP = (ADDR>=^h5000)&(ADDR<^h6000) & TM
ERR_LOG = (ADDR>=^h6000)&(ADDR<^h7000) & TM

Check address modifier, Use Fermilab requested
09, 0A, 0B, and HUHEPL favorites 29, 2D, 39 & 3D.

AM[] = 5:0
%AMOK% = ( (AM=^h09) # (AM=^h0A) # (AM=^h0B)
# (AM=^h29) # (AM=^h2D) # (AM=^h39) # (AM=^h3D) )

%D0_IT% = ( TM & !ACK & !WORD )

%ADDR_VALID% = ( (CODE # MASK # MCLK # MOM # RSM # FIB_INT# TAXI #
PIPE_CAP # ERR_LOG) & (!WORD) & (!ACK) )

If board is decoded then enable the VME data bus
and start the delay line on either STROBE

DEL = (DO_IT) & (S0 # S1) & (AMOK)

ENDAT = (DO_IT) & (LADDR) & (AMOK)

If board is decoded, bang the appropriate STROBE
lines

STROBE0 = (ADDR_VALID) & (AMOK) &
((LADDR & !WRITE) # (DEL4 & !DEL5 & WRITE))

STROBE1 = (ADDR_VALID) & (AMOK) & S1 &
((LADDR & !WRITE) # (DEL4 & !DEL5 & WRITE))

If the address has responded by DEL5, then ACKNOWLEDGE,
otherwise hit the electric dog collar button!

CLK_ACK = (OK#CODE#MASK) & DEL5
& (ADDR_VALID)

CLK_ERR = DEL5 & !(OK#CODE#MASK)

Bye now!
```

## 2.6 HDL Code

### 2.6.1 VME\_INT

```
" D:\CDF\SRC\VME\VME_INT.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.02c0
" Mon Mar 17 09:33:26 1997

" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are speed optimized.
```

```
MODULE VME_INT
```

#### DECLARATIONS

"Input variables

```
ADDR0 PIN;
ADDR1 PIN;
ADDR2 PIN;
ADDR3 PIN;
ADDR4 PIN;
ADDR5 PIN;
ADDR6 PIN;
ADDR7 PIN;
ADDR8 PIN;
ADDR9 PIN;
ADDR10 PIN;
ADDR11 PIN;
ADDR12 PIN;
ADDR13 PIN;
ADDR14 PIN;
AM0 PIN;
AM1 PIN;
AM2 PIN;
AM3 PIN;
AM4 PIN;
AM5 PIN;
DEL4 PIN;
DEL5 PIN;
IACK PIN;
LADDR PIN;
LWORD PIN;
OK PIN;
S0 PIN;
S1 PIN;
TM PIN;
WRITE PIN;
```

"Logic variables

```

CLK_ACK PIN ISTYPE 'com';
CLK_ERR PIN ISTYPE 'com';
CODE PIN ISTYPE 'com';
DEL PIN ISTYPE 'com';
ENDAT PIN ISTYPE 'com';
ERR_LOG PIN ISTYPE 'com';
FIB_INT PIN ISTYPE 'com';
MASK PIN ISTYPE 'com';
MCLK PIN ISTYPE 'com';
MCM PIN ISTYPE 'com';
PAD PIN ISTYPE 'com';
PIPE_CAP PIN ISTYPE 'com';
RSM PIN ISTYPE 'com';
STROBE0 PIN ISTYPE 'com';
STROBE1 PIN ISTYPE 'com';
TAXI PIN ISTYPE 'com';

"Vectors
DECLARATIONS
    ADDR=[  

        ADDR14,  

        ADDR13,  

        ADDR12,  

        ADDR11,  

        ADDR10,  

        ADDR9,  

        ADDR8,  

        ADDR7,  

        ADDR6,  

        ADDR5,  

        ADDR4,  

        ADDR3,  

        ADDR2,  

        ADDR1,  

        ADDR0  

    ];  

    AM=[  

        AM5,  

        AM4,  

        AM3,  

        AM2,  

        AM1,  

        AM0  

    ];  

"Logic Equations
EQUATIONS
    CLK_ACK = !IACK & !LWORD & ERR_LOG & DEL5 & OK # !IACK & !LWORD & PIPE_CAP  

        & DEL5 & OK # !IACK & !LWORD & TAXI & DEL5 & OK # !IACK & !LWORD &  

        FIB_INT & DEL5 & OK # !IACK & !LWORD & RSM & DEL5 & OK # !IACK & !LWORD  

&  

        MCM & DEL5 & OK # !IACK & !LWORD & MCLK & DEL5 & OK # !IACK & !LWORD  

&  

        CODE & DEL5 # !IACK & !LWORD & MASK & DEL5 ;

```

```

CLK_ERR = !OK & !CODE & !MASK & DEL5 ;

CODE = !WRITE & TM & !ADDR14 & !ADDR13 & !ADDR12 & !ADDR11 & !ADDR10 & !
       ADDR9 & !ADDR8 & !ADDR7 & !ADDR6 & !ADDR5 & !ADDR4 & !ADDR3 & !ADDR2 &
!ADDR1
       & !ADDR0 ;

DEL = AM0 & !AM1 & AM3 & AM5 & S0 & TM & !IACK & !LWORD # AM1 & !AM2 & AM3
       & !AM4 & !AM5 & S0 & TM & !IACK & !LWORD # AM0 & !AM1 & !AM2 & AM3 &
!AM4
       & S0 & TM & !IACK & !LWORD # AM0 & !AM1 & AM3 & AM5 & S1 & TM & !IACK & !
       LWORD # AM1 & !AM2 & AM3 & !AM4 & !AM5 & S1 & TM & !IACK & !LWORD #
AM0 &
       !AM1 & !AM2 & AM3 & !AM4 & S1 & TM & !IACK & !LWORD ;

ENDAT = AM0 & !AM1 & AM3 & AM5 & LADDR & TM & !IACK & !LWORD # AM1 & !AM2
       & AM3 & !AM4 & !AM5 & LADDR & TM & !IACK & !LWORD # AM0 & !AM1 & !AM2 &
AM3
       & !AM4 & LADDR & TM & !IACK & !LWORD ;

ERR_LOG = TM & !ADDR12 & ADDR13 & ADDR14 ;

FIB_INT = TM & !ADDR14 & ADDR12 & ADDR13 ;

MASK = TM & !ADDR14 & !ADDR13 & !ADDR12 & !ADDR11 & !ADDR10 & !ADDR9 & !
       ADDR8 & !ADDR7 & !ADDR6 & !ADDR5 & !ADDR4 & !ADDR3 & ADDR2 & !ADDR1 &
!ADDR0
       ;
MCLK = TM & !ADDR8 & !ADDR9 & !ADDR10 & !ADDR11 & !ADDR13 & !ADDR14 & ADDR12
       ;
MCM = TM & !ADDR9 & !ADDR10 & !ADDR11 & !ADDR13 & !ADDR14 & ADDR8 & ADDR12
       # TM & !ADDR0 & !ADDR6 & !ADDR7 & !ADDR8 & !ADDR10 & !ADDR11 & !ADDR13 &
!
       ADDR14 & ADDR9 & ADDR12 # TM & !ADDR1 & !ADDR6 & !ADDR7 & !ADDR8 &
!ADDR10
       & !ADDR11 & !ADDR13 & !ADDR14 & ADDR9 & ADDR12 # TM & !ADDR2 & !ADDR6 &
!
       ADDR7 & !ADDR8 & !ADDR10 & !ADDR11 & !ADDR13 & !ADDR14 & ADDR9 & ADDR12
#
       TM & !ADDR3 & !ADDR6 & !ADDR7 & !ADDR8 & !ADDR10 & !ADDR11 & !ADDR13 & !
       ADDR14 & ADDR9 & ADDR12 # TM & !ADDR4 & !ADDR6 & !ADDR7 & !ADDR8 &
!ADDR10
       & !ADDR11 & !ADDR13 & !ADDR14 & ADDR9 & ADDR12 # TM & !ADDR5 & !ADDR6 &
!
       ADDR7 & !ADDR8 & !ADDR10 & !ADDR11 & !ADDR13 & !ADDR14 & ADDR9 & ADDR12
;
PAD = !ADDR14 & !ADDR13 & !ADDR12 & !ADDR11 & !ADDR10 & !ADDR9 & !ADDR8 & !
       ADDR7 & !ADDR6 & !ADDR5 & !ADDR4 & !ADDR3 & !ADDR1 & !ADDR0 & !WRITE &
TM ;

PIPE_CAP = TM & !ADDR13 & ADDR12 & ADDR14 ;

```

```

RSM = TM & !ADDR12 & !ADDR14 & ADDR13 ;

STROBE0 = DEL4 & !DEL5 & WRITE & AM0 & !AM1 & !AM2 & AM3 & !AM4 & !IACK & !
LWORD & CODE # LADDR & !WRITE & AM0 & !AM1 & !AM2 & AM3 & !AM4 & !IACK
& !
LWORD & CODE # DEL4 & !DEL5 & WRITE & AM1 & !AM2 & AM3 & !AM4 & !AM5 & !
IACK & !LWORD & CODE # LADDR & !WRITE & AM1 & !AM2 & AM3 & !AM4 & !AM5
& !
IACK & !LWORD & CODE # DEL4 & !DEL5 & WRITE & AM0 & !AM1 & AM3 & AM5 & !
IACK & !LWORD & CODE # LADDR & !WRITE & AM0 & !AM1 & AM3 & AM5 & !IACK
& !
LWORD & CODE # DEL4 & !DEL5 & WRITE & AM0 & !AM1 & !AM2 & AM3 & !AM4 & !
IACK & !LWORD & MASK # LADDR & !WRITE & AM0 & !AM1 & !AM2 & AM3 & !AM4
& !
IACK & !LWORD & MASK # DEL4 & !DEL5 & WRITE & AM1 & !AM2 & AM3 & !AM4 &
!
AM5 & !IACK & !LWORD & MASK # LADDR & !WRITE & AM1 & !AM2 & AM3 & !AM4
& !
AM5 & !IACK & !LWORD & MASK # DEL4 & !DEL5 & WRITE & AM0 & !AM1 & AM3 &
AM5 & !IACK & !LWORD & MASK # LADDR & !WRITE & AM0 & !AM1 & AM3 & AM5 &
!
IACK & !LWORD & MCLK # LADDR & !WRITE & AM0 & !AM1 & !AM2 & AM3 & !AM4
& !
IACK & !LWORD & MCLK # DEL4 & !DEL5 & WRITE & AM1 & !AM2 & AM3 & !AM4 &
!
AM5 & !IACK & !LWORD & MCLK # LADDR & !WRITE & AM1 & !AM2 & AM3 & !AM4
& !
AM5 & !IACK & !LWORD & MCLK # DEL4 & !DEL5 & WRITE & AM0 & !AM1 & AM3 &
AM5 & !IACK & !LWORD & MCLK # LADDR & !WRITE & AM0 & !AM1 & AM3 & AM5 &
!
IACK & !LWORD & MCM # LADDR & !WRITE & AM0 & !AM1 & !AM2 & AM3 & !AM4 &
!
IACK & !LWORD & MCM # DEL4 & !DEL5 & WRITE & AM1 & !AM2 & AM3 & !AM4 &
!
AM5 & !IACK & !LWORD & MCM # LADDR & !WRITE & AM1 & !AM2 & AM3 & !AM4 &
!
IACK & !LWORD & MCM # DEL4 & !DEL5 & WRITE & AM0 & !AM1 & AM3 & AM5 & !IACK
& !
IACK & !LWORD & MCM # LADDR & !WRITE & AM0 & !AM1 & AM3 & AM5 & !IACK
& !
LWORD & MCM # DEL4 & !DEL5 & WRITE & AM0 & !AM1 & !AM2 & AM3 & !AM4 &
!
IACK & !LWORD & RSM # LADDR & !WRITE & AM0 & !AM1 & !AM2 & AM3 & !AM4 &
!
IACK & !LWORD & RSM # DEL4 & !DEL5 & WRITE & AM1 & !AM2 & AM3 & !AM4 & !AM5 & !
IACK & !LWORD & RSM # LADDR & !WRITE & AM1 & !AM2 & AM3 & !AM4 & !AM5 &
!
```





```

# LADDR & !WRITE & S1 & AM1 & !AM2 & AM3 & !AM4 & !AM5 & !IACK & !LWORD
&
MCM # DEL4 & !DEL5 & WRITE & S1 & AM0 & !AM1 & !AM2 & AM3 & !AM4 & !IACK
&
!LWORD & MCM # LADDR & !WRITE & S1 & AM0 & !AM1 & !AM2 & AM3 & !AM4 &
!IACK
& !LWORD & MCM # DEL4 & !DEL5 & WRITE & S1 & AM0 & !AM1 & AM3 & AM5 & !
IACK & !LWORD & MCLK # LADDR & !WRITE & S1 & AM0 & !AM1 & AM3 & AM5 &
!IACK
& !LWORD & MCLK # DEL4 & !DEL5 & WRITE & S1 & AM1 & !AM2 & AM3 & !AM4 &
!
AM5 & !IACK & !LWORD & MCLK # LADDR & !WRITE & S1 & AM1 & !AM2 & AM3 &
!AM4
& !AM5 & !IACK & !LWORD & MCLK # DEL4 & !DEL5 & WRITE & S1 & AM0 & !AM1
&
!AM2 & AM3 & !AM4 & !IACK & !LWORD & MCLK # LADDR & !WRITE & S1 & AM0 &
!
AM1 & !AM2 & AM3 & !AM4 & !IACK & !LWORD & MCLK # DEL4 & !DEL5 & WRITE
& S1
& AM0 & !AM1 & AM3 & AM5 & !IACK & !LWORD & MASK # DEL4 & !DEL5 & WRITE &
S1
& AM1 & !AM2 & AM3 & !AM4 & !AM5 & !IACK & !LWORD & MASK # LADDR &
!WRITE &
S1 & AM1 & !AM2 & AM3 & !AM4 & !AM5 & !IACK & !LWORD & MASK # DEL4 &
!DEL5
& WRITE & S1 & AM0 & !AM1 & !AM2 & AM3 & !AM4 & !IACK & !LWORD & MASK #
LADDR & !WRITE & S1 & AM0 & !AM1 & !AM2 & AM3 & !AM4 & !IACK & !LWORD &
MASK
# DEL4 & !DEL5 & WRITE & S1 & AM0 & !AM1 & AM3 & AM5 & !IACK & !LWORD &
CODE # LADDR & !WRITE & S1 & AM0 & !AM1 & AM3 & AM5 & !IACK & !LWORD &
CODE
# DEL4 & !DEL5 & WRITE & S1 & AM1 & !AM2 & AM3 & !AM4 & !AM5 & !IACK &
!IACK
& !LWORD & CODE # DEL4 & !DEL5 & WRITE & S1 & AM0 & !AM1 & !AM2 & AM3 &
!
AM4 & !IACK & !LWORD & CODE # LADDR & !WRITE & S1 & AM0 & !AM1 & !AM2 &
AM3
& !AM4 & !IACK & !LWORD & CODE ;
TAXI = TM & !ADDR12 & !ADDR13 & ADDR14 ;
END VME_INT

```





### 3. Master Clock

#### 3.1 Pin Description

Design				CDF/SRC/MASTCLK	
Date				14-Nov-96	
				3/20/98 14:16	
Signal	I	O	OT	I/O	Function
MCMAC[7:0]		8			Master Clock Memory Address lines
MCDAT[7:0]				8	Master Clock Memory Data lines
/MCOE		1			MC Memory Output Enable
/MCWE		1			MC Memory Write Enable
/MCM	1				MC Memory VME Request line
/MCLK	1				VME Request line (Control/Status)
/STROBE0	1				VME Data Strobe
/VMEOK		1			VME Local Acknowledge
/BWRITE	1				VME board Write line
VDAT[7:0]				8	VME board data lines
VADDR[8:1]	8				VME Address lines
BCN[7:0]		8			Bunch Counter
TURN[15:0]		16			Turn Counter
ADV_PIPE		1			Advance pipeline output to FIB
VBX		1			Possible beam location in 400ns mode
SYNC		1			Sync output (Phase 0 & 1)
BXING		1			Bunch Crossing output
BZERO		1			Bunch Zero output
GAP		1			Gap output
/LASER		1			Trigger output to external laser
/EN_CDF_CLK		1			Enable external Master Clock receivers
CDF_CLK[6:0]	7				Inputs from CDF Master Clock
/SCINT[2:0]	3				3x Trigger scintillator inputs
EXT_TRIG		1			External Trigger output to TAXI
TAXI_CLK		1			15.15MHz Clock to Taxi Chip
RF_IN	1				53 MHz Clock input
RF_OUT		1			53 MHz Clock output
RSM_CLK		1			132ns clock for RSM FPGA
RSM_CLK_IN	1				132ns clock for counters & SMs
VCO		1			53.104 MHz clock input from PLL crystal
UP			1		Phase detector output
!UP			1		Phase detector output
DN			1		Phase detector output
!DN			1		Phase detector output
/LOCK	1				PLL Lock input from comparator
/LOCK_IND		1			PLL Lock indicator
/MODE132_IND		1			132ns Mode indicator
/MODE400_IND		1			400ns Mode indicator
/CONFIG1		1			Local configuration indicator
<b>Pin Count</b>	<b>25</b>	<b>52</b>	<b>4</b>	<b>16</b>	<b>97</b>

### 3.2 Constraint File

```
# Design: Mastclk
# Created by XACT Floorplanner ver 6.0.1
NOTPLACE BLOCK *: N2 P5 N6 N7 N8 A9 C13 G14;
place block ADV_PIPE      : D14 ;
place block N_EN_RSM     : A13 ;
place block N_XQT        : A11 ;
place block BCN0          : N11 ;
place block BCN1          : M10 ;
place block BCN2          : P9 ;
place block BCN3          : P12 ;
place block BCN4          : P11 ;
place block BCN5          : N10 ;
place block BCN6          : P10 ;
place block BCN7          : N9 ;
place block BXING_        : K13 ;
place block BZERO_        : J14 ;
place block CDF_CLK0      : A7 ;
place block CDF_CLK1      : A8 ;
place block CDF_CLK2      : H1 ;
place block CDF_CLK3      : G2 ;
place block DOWN          : C2 ;
place block EXT_TRIG      : M9 ;
place block GAP_          : J2 ;
place block MCDAT0        : G1 ;
place block MCDAT1        : F1 ;
place block MCDAT2        : F2 ;
place block MCDAT3        : E1 ;
place block MCDAT4        : F3 ;
place block MCDAT5        : E2 ;
place block MCDAT6        : D2 ;
place block MCDAT7        : E3 ;
place block MCMAC0        : N1 ;
place block MCMAC1        : M1 ;
place block MCMAC2        : K3 ;
place block MCMAC3        : L2 ;
place block MCMAC4        : L1 ;
place block MCMAC5        : K2 ;
place block MCMAC6        : J3 ;
place block MCMAC7        : K1 ;
place block N_BWRITE       : J1 ;
place block N_CONFIG1      : C10 ;
place block N_EN_CDF_CLK   : C12 ;
place block N_LOCK         : C9 ;
place block N_LOCK_IND     : B9 ;
place block N_MCLK         : A6 ;
place block N_MCM          : A2 ;
place block N_MCOE         : D1 ;
place block N_MCWE         : A10 ;
place block N_MODE132_IND   : F13 ;
place block N_MODE400_IND   : B8 ;
place block N_SCINT0        : P7 ;
place block N_SCINT1        : P6 ;
place block N_SCINT2        : P8 ;
```

```

place block N_STROBE0      : B7 ;
place block N_VMEOK1       : H2 ;
place block RF_IN          : C1 ;
place block RF_OUT         : B1 ;
place block RSM_CLK        : B10 ;
place block SYNC            : E12 ;
place block TAXI_CLK       : E13 ;
place block TURN0           : F12 ;
place block TURN1           : F14 ;
place block TURN2           : H14 ;
place block TURN3           : G13 ;
place block TURN4           : K14 ;
place block TURN5           : J13 ;
place block TURN6           : J12 ;
place block TURN7           : H13 ;
place block TURN8           : K12 ;
place block TURN9           : L13 ;
place block TURN10          : L14 ;
place block TURN11          : N14 ;
place block TURN12          : M13 ;
place block TURN13          : M14 ;
place block TURN14          : N12 ;
place block TURN15          : M12 ;
place block UP               : B3 ;
place block UP_B             : B2 ;
place block VADDR1          : N3 ;
place block VADDR2          : P2 ;
place block VADDR3          : N4 ;
place block VADDR4          : M5 ;
place block VADDR5          : P3 ;
place block VADDR6          : P4 ;
place block VADDR7          : N5 ;
place block VADDR8          : M6 ;
place block VBX              : E14 ;
place block VCO              : C3 ;
place block VDAT0            : B6 ;
place block VDAT1            : A5 ;
place block VDAT2            : C6 ;
place block VDAT3            : B5 ;
place block VDAT4            : A4 ;
place block VDAT5            : A3 ;
place block VDAT6            : C5 ;
place block VDAT7            : B4 ;

```

```

PLACE BLOCK WINDOW/CRW/STATE3: [GF KK];
PLACE BLOCK WINDOW/CRW/STATE2: [GF KK];
PLACE BLOCK WINDOW/CRW/STATE1: [GF KK];
PLACE BLOCK WINDOW/CRD/SV1: [GF KK];
PLACE BLOCK WINDOW/CRD/SV0: [GF KK];
PLACE BLOCK WINDOW/$1N52: [GF KK];
PLACE BLOCK WINDOW/WW4: [GF KK];
PLACE BLOCK WINDOW/WW2: [GF KK];
PLACE BLOCK WINDOW/WW0: [GF KK];
PLACE BLOCK WINDOW/DD4: [GF KK];
PLACE BLOCK WINDOW/DD2: [GF KK];
PLACE BLOCK WINDOW/DD0: [GF KK];

```

```

PLACE BLOCK WINDOW/CRW/_23_: [GF KK];
PLACE BLOCK WINDOW/CRW/STATE0: [GF KK];
PLACE BLOCK TRIG: [GF KK];
PLACE BLOCK WINDOW/$1N64: [GF KK];
PLACE BLOCK WINDOW/SCINT_OUT: [GF KK];
PLACE BLOCK WINDOW/$1N54: [GF KK];
PLACE BLOCK WINDOW/WCOMP/$1N68: [GF KK];
PLACE BLOCK WINDOW/WCOMP/$1N67: [GF KK];
PLACE BLOCK WINDOW/$1N68: [GF KK];
PLACE BLOCK WINDOW/DCOMP/$1N68: [GF KK];
PLACE BLOCK WINDOW/DCOMP/$1N67: [GF KK];
PLACE BLOCK WINDOW/$1N58: [GF KK];
PLACE BLOCK VIRTUALBX: GE;
PLACE BLOCK MODE132: JE;
PLACE BLOCK TRN15: OM;
PLACE BLOCK TRN14: OL;
PLACE BLOCK TRN12: PM;
PLACE BLOCK TRN10: PK;
PLACE BLOCK TRN8: OK;
PLACE BLOCK TCU/T4: PL;
PLACE BLOCK TRN7: PH;
PLACE BLOCK TRN6: OI;
PLACE BLOCK TRN4: PI;
PLACE BLOCK TRN2: OG;
PLACE BLOCK TRN0: PG;
PLACE BLOCK TCL/T4: OH;
PLACE BLOCK PHASE_DET/NODE4: AA;
PLACE BLOCK PHASE_DET/NODE1: AB;
PLACE BLOCK PHASE_DET/$1N89: BB;
PLACE BLOCK $1N519: AC;
PLACE BLOCK $1N515: BA;
PLACE BLOCK MAC/MEMCLR: [BK DN];
PLACE BLOCK MAC/$1N25: [BK DN];
PLACE BLOCK NGTD6: [BK DN];
PLACE BLOCK NGTD5: [BK DN];
PLACE BLOCK NGTD4: [BK DN];
PLACE BLOCK NGTD2: [BK DN];
PLACE BLOCK NGTD1: [BK DN];
PLACE BLOCK MAC/$1I10/T6: [BK DN];
PLACE BLOCK MAC/$1I10/T3: [BK DN];
PLACE BLOCK MACADDR7: AH;
PLACE BLOCK MACADDR6: AI;
PLACE BLOCK MACADDR5: AJ;
PLACE BLOCK MACADDR4: AK;
PLACE BLOCK MACADDR3: AL;
PLACE BLOCK MACADDR2: AM;
PLACE BLOCK MACADDR1: AN;
PLACE BLOCK R_WIDTH/$1I37: TBUF.EB.2;
PLACE BLOCK R_WIDTH/$1I33: TBUF.FB.1;
PLACE BLOCK R_WIDTH/$1I32: TBUF.FB.2;
PLACE BLOCK R_WIDTH/$1I31: TBUF.GB.1;
PLACE BLOCK R_WIDTH/$1I30: TBUF.GB.2;
PLACE BLOCK BUNCH7: MN;
PLACE BLOCK BUNCH6: MK;
PLACE BLOCK BUNCH4: NK;

```

```
PLACE BLOCK BUNCH2: NM;  
PLACE BLOCK BUNCH0: NL;  
PLACE BLOCK BC/TC: ML;  
PLACE BLOCK BC/T4: MM;  
PLACE BLOCK $2N181: NN;  
# End
```

### **3.3 Placement Report**

Error! Not a valid filename.

### **3.4 XACT Performance**

Error! Not a valid filename.

### 3.5 State CAD Diagrams

#### 3.5.1 VME

SRC/MASTCLK/VME  
 VME Decoder  
 1 Aug., 1996  
 Updated 1 Mar, 1997

ADDR[] = 3:0

WCR = MCLK &  
 ADDR[0] & WRITE

WWID = MCLK &  
 ADDR[4] & WRITE  
 & !ENMCLK

RCR = MCLK &  
 ADDR[0] & !WRITE

RWID = MCLK &  
 ADDR[4] & !WRITE

RSR = MCLK &  
 ADDR[1] & !WRITE

WMCM = MCM &  
 WRITE & !ENMCLK

WMASK = MCLK &  
 ADDR[2] & WRITE  
 & !ENMCLK

RMCM = MCM &  
 !WRITE & !ENMCLK

RMASK = MCLK &  
 ADDR[2] & !WRITE

XQOK = MCLK &  
 ADDR[8] & WRITE  
 & ENMCLK

WDEL = MCLK &  
 ADDR[3] & WRITE  
 & !ENMCLK

RDEL = MCLK &  
 ADDR[3] & !WRITE

---

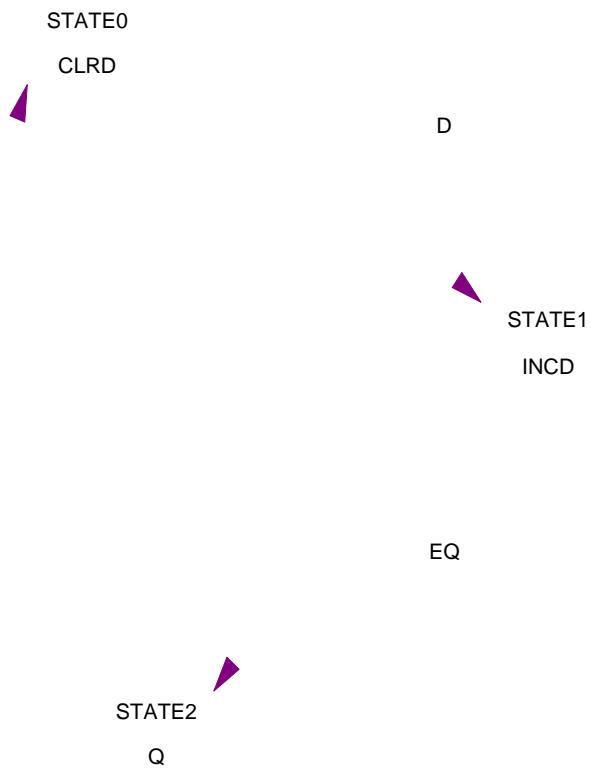
VMEOK = MCLK & WRITE & (ADDR[0] # ADDR[1] # (ENMCLK & ADDR[8])  
 ) #  
 (!ENMCLK & (ADDR[2] # ADDR[3] # ADDR[4]))  
 #  
 MCLK & !WRITE & (ADDR[0] # ADDR[1]  
 # ADDR[2] # ADDR[3] # ADDR[4] # ADDR[8])  
 #  
 MCM & !ENMCLK

RDVME = !WRITE & (MCLK # MCM)

WRTVME = WRITE & (MCLK # MCM)

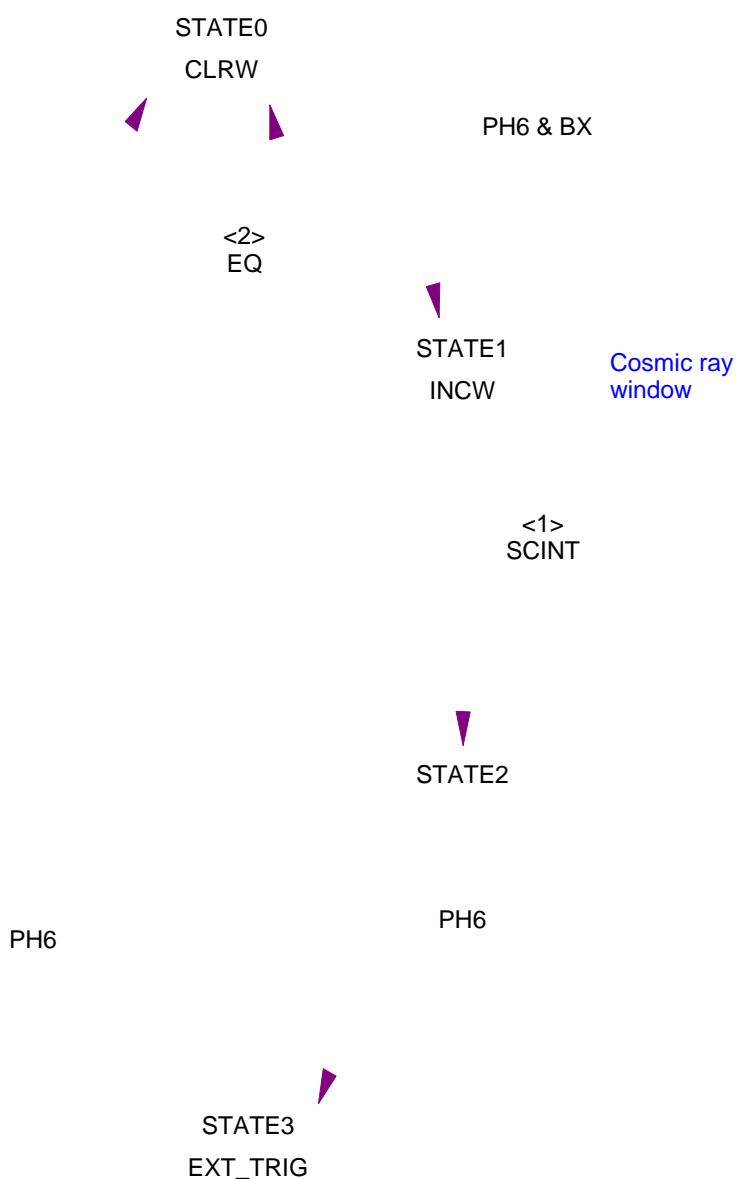
### 3.5.2 CRD\_REV1

CRD\_REV1  
Cosmic ray delay  
Aug 2, 1996

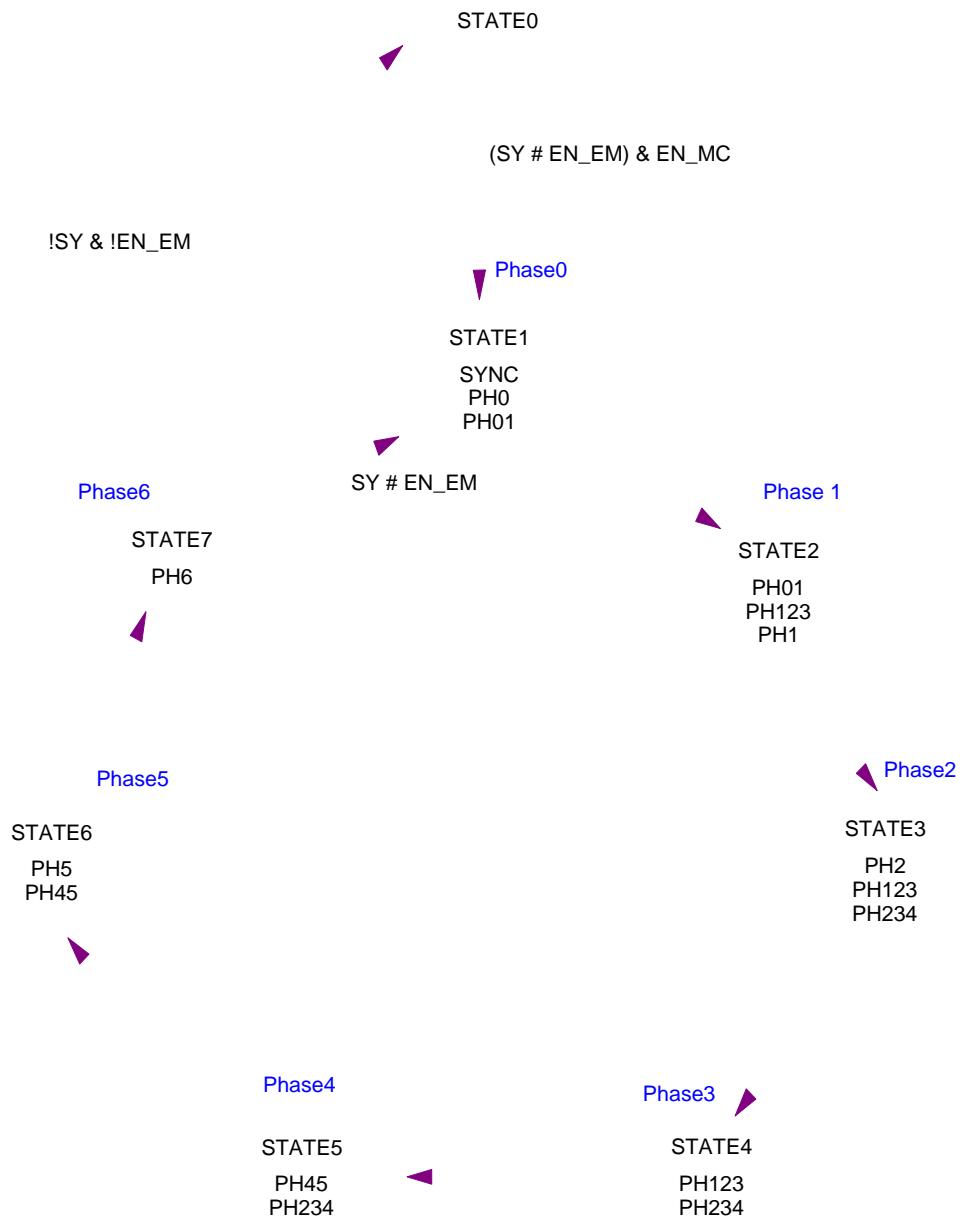


### 3.5.3 CRW\_REV1

CRW\_REV1  
Cosmic ray window  
Aug 2, 1996 JO

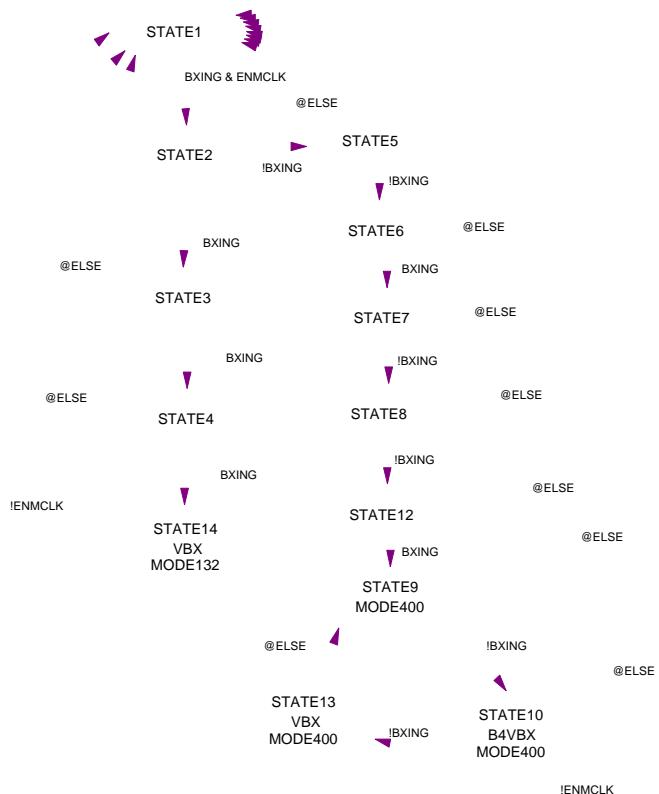


### 3.5.4 DIV7\_REV1



### 3.5.5 VBX\_REV1

SRC/VBX  
Detects 132 or 400 modes  
March 13, 1997



### **3.6 HDL Code**

#### **3.6.1 VME**

```
" D:\CDF\SRC\MASTCLK\VME.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.02c0
" Wed Mar 12 23:26:59 1997
```

```
" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are speed optimized.
```

```
MODULE VME
```

```
DECLARATIONS
```

```
"Input variables
```

```
    addr0 PIN;
    addr1 PIN;
    addr2 PIN;
    ADDR3 PIN;
    ENMCLK PIN;
    MCLK PIN;
    MCM PIN;
    WRITE PIN;
```

```
"Logic variables
```

```
    RCR PIN ISTYPE 'com';
    RDEL PIN ISTYPE 'com';
    RDVME PIN ISTYPE 'com';
    RMASK PIN ISTYPE 'com';
    RMCM PIN ISTYPE 'com';
    RSR PIN ISTYPE 'com';
    RWID PIN ISTYPE 'com';
    VMEOK PIN ISTYPE 'com';
    WCR PIN ISTYPE 'com';
    WDEL PIN ISTYPE 'com';
    WMASK PIN ISTYPE 'com';
    WMCM PIN ISTYPE 'com';
    WRTVME PIN ISTYPE 'com';
    WWID PIN ISTYPE 'com';
    XQOK PIN ISTYPE 'com';
```

```
"Vectors
```

```
DECLARATIONS
```

```
    ADDR=[  
        ADDR3,  
        addr2,  
        addr1,  
        addr0  
    ];
```

```

"Logic Equations
EQUATIONS
RCR = !WRITE & !ADDR3 & !addr2 & !addr1 & !addr0 & MCLK ;
RDEL = !WRITE & !ADDR3 & !addr2 & addr1 & addr0 & MCLK ;
RDVME = MCM & !WRITE # MCLK & !WRITE ;
RMASK = !WRITE & !ADDR3 & !addr2 & addr1 & !addr0 & MCLK ;
RMCM = MCM & !WRITE & !ENMCLK ;
RSR = !WRITE & !ADDR3 & !addr2 & !addr1 & addr0 & MCLK ;
RWID = !WRITE & !ADDR3 & addr2 & !addr1 & !addr0 & MCLK ;
VMEOK = !ADDR3 & !addr2 & !ENMCLK & MCLK # !ADDR3 & !addr1 & !addr0 & !
      ENMCLK & MCLK # !addr2 & !addr1 & !addr0 & ENMCLK & MCLK # !ADDR3 & !
      addr2 & !addr1 & MCLK # !ADDR3 & !addr1 & !addr0 & MCLK & !WRITE # !ADDR3
      & !addr2 & MCLK & !WRITE # !addr2 & !addr1 & !addr0 & MCLK & !WRITE #
      MCM & !ENMCLK ;
WCR = WRITE & !ADDR3 & !addr2 & !addr1 & !addr0 & MCLK ;
WDEL = WRITE & !ENMCLK & !ADDR3 & !addr2 & addr1 & addr0 & MCLK ;
WMASK = WRITE & !ENMCLK & !ADDR3 & !addr2 & addr1 & !addr0 & MCLK ;
WMCM = MCM & WRITE & !ENMCLK ;
WRTVME = MCM & WRITE # MCLK & WRITE ;
WWID = WRITE & !ENMCLK & !ADDR3 & addr2 & !addr1 & !addr0 & MCLK ;
XQOK = WRITE & ENMCLK & ADDR3 & !addr2 & !addr1 & !addr0 & MCLK ;
END VME

```

### 3.6.2 CRD\_REV1

```
" D:\CDF\SRC\MASTCLK\CRD_REV1.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.00
" Fri Aug 02 11:48:26 1996
```

```
MODULE CRD_REV1
```

#### DECLARATIONS

```
"clock name
    CLK PIN;
```

```
"Input variables
```

```
    D PIN;
    EQ PIN;
```

```
"Output variables
```

```
    CLRD PIN ISTYPE 'reg';
    INCD PIN ISTYPE 'reg';
    Q PIN ISTYPE 'reg';
```

```
"State variables
```

```
    SV0 PIN ISTYPE 'reg';
    SV1 PIN ISTYPE 'reg';
```

```
"Clocked logic clock setup
```

#### EQUATIONS

```
    CLRD.clk=CLK;
    INCD.clk=CLK;
    Q.clk=CLK;
```

```
"State Register assignment
```

#### DECLARATIONS

```
    sreg=[ SV0,SV1];
```

#### EQUATIONS

```
    sreg.clk=CLK;
```

#### DECLARATIONS

```
    STATE0=[0, 0];
    STATE1=[0, 1];
    STATE2=[1, 0];
```

#### EQUATIONS

```
    SV0 := EQ & !SV0.FB & SV1.FB ;
```

```
    SV1 := D & !SV0.FB & !SV1.FB # !EQ & !SV0.FB & SV1.FB ;
```

```
    CLRD := !D & !SV1.FB # SV0.FB & !SV1.FB ;
```

```
INCD := D & !SV0.FB & !SV1.FB # !EQ & !SV0.FB & SV1.FB ;
```

```
Q := EQ & !SV0.FB & SV1.FB ;  
END CRD_REV1
```

### 3.6.3 CRW\_REV1

```
" D:\CDF\SRC\MASTCLK\CRW_REV1.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.02c0
" Wed Jun 04 23:07:00 1997

" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are speed optimized.

MODULE CRW_REV1

DECLARATIONS
"clock name
    CLK PIN;

"Input variables
    BX PIN;
    EQ PIN;
    PH6 PIN;
    SCINT PIN;

"Output variables
    CLRW PIN ISTYPE 'reg';
    EXT_TRIG PIN ISTYPE 'reg';
    INCW PIN ISTYPE 'reg';

"State variables
    SV0 PIN ISTYPE 'reg';
    SV1 PIN ISTYPE 'reg';

"EQUATIONS
    Clocked logic clock setup
EQUATIONS
    CLRW.clk=CLK;
    EXT_TRIG.clk=CLK;
    INCW.clk=CLK;

"State Register assignment
DECLARATIONS
    sreg=[ SV0,SV1];

EQUATIONS
    sreg.clk=CLK;

DECLARATIONS
    STATE0=[0, 0];
    STATE1=[0, 1];
    STATE2=[1, 0];
    STATE3=[1, 1];
```

## EQUATIONS

```
SV0 := SCINT & !SV0.FB & SV1.FB # SV0.FB & !SV1.FB # !PH6 & SV0.FB ;  
SV1 := PH6 & BX & !SV1.FB # !EQ & !SCINT & !SV0.FB & SV1.FB # PH6 &  
      SV0.FB & !SV1.FB # !PH6 & SV0.FB & SV1.FB ;  
  
CLRW := !PH6 & !SV0.FB & !SV1.FB # !BX & !SV0.FB & !SV1.FB # !SCINT & EQ  
      & !SV0.FB & SV1.FB # PH6 & SV0.FB & SV1.FB ;  
EXT_TRIG := PH6 & SV0.FB & !SV1.FB # !PH6 & SV0.FB & SV1.FB ;  
INCW := PH6 & BX & !SV0.FB & !SV1.FB # !EQ & !SCINT & !SV0.FB & SV1.FB ;  
END CRW_REV1
```

### 3.6.4 DIV7\_REV1

```
" D:\CDF\SRC\MASTCLK\DIV7REV1.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.02c0
" Tue Jun 03 01:29:53 1997
```

```
" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are speed optimized.
```

```
MODULE DIV7REV1
```

#### DECLARATIONS

```
"clock name
    CLK PIN;
```

```
"Input variables
```

```
    EN_EM PIN;
    EN_MC PIN;
    SY PIN;
```

```
"Output variables
```

```
    PH0 PIN ISTYPE 'reg';
    PH01 PIN ISTYPE 'reg';
    PH1 PIN ISTYPE 'reg';
    PH2 PIN ISTYPE 'reg';
    PH5 PIN ISTYPE 'reg';
    PH6 PIN ISTYPE 'reg';
    PH45 PIN ISTYPE 'reg';
    PH123 PIN ISTYPE 'reg';
    PH234 PIN ISTYPE 'reg';
    SYNC PIN ISTYPE 'reg';
```

```
"State variables
```

```
    SV0 PIN ISTYPE 'reg';
    SV1 PIN ISTYPE 'reg';
    SV2 PIN ISTYPE 'reg';
```

```
"Clocked logic clock setup
```

#### EQUATIONS

```
    PH0.clk=CLK;
    PH01.clk=CLK;
    PH1.clk=CLK;
    PH2.clk=CLK;
    PH5.clk=CLK;
    PH6.clk=CLK;
    PH45.clk=CLK;
    PH123.clk=CLK;
    PH234.clk=CLK;
    SYNC.clk=CLK;
```

```

"State Register assignment
DECLARATIONS
    sreg=[ SV0,SV1,SV2];

EQUATIONS
    sreg.clk=CLK;

DECLARATIONS
    STATE0=[0, 0, 0];
    STATE1=[0, 0, 1];
    STATE2=[0, 1, 0];
    STATE3=[0, 1, 1];
    STATE4=[1, 0, 0];
    STATE5=[1, 0, 1];
    STATE6=[1, 1, 0];
    STATE7=[1, 1, 1];

EQUATIONS
    SV0 := !SV0.FB & SV1.FB & SV2.FB # SV0.FB & !SV1.FB # SV0.FB & !SV2.FB ;
    SV1 := !SV1.FB & SV2.FB # SV1.FB & !SV2.FB ;
    SV2 := EN_MC & SY & !SV2.FB # EN_MC & EN_EM & !SV2.FB # EN_EM & SV0.FB &
        SV1.FB # SY & SV0.FB & SV1.FB # SV0.FB & !SV2.FB # SV1.FB & !SV2.FB ;
    PH0 := EN_MC & SY & !SV0.FB & !SV1.FB & !SV2.FB # EN_MC & EN_EM & !SV0.FB &
        !SV1.FB & !SV2.FB # EN_EM & SV0.FB & SV1.FB & SV2.FB # SY & SV0.FB &
        SV1.FB & SV2.FB ;
    PH01 := EN_MC & SY & !SV0.FB & !SV1.FB # EN_MC & EN_EM & !SV0.FB & !SV1.FB
        # EN_EM & SV0.FB & SV1.FB & SV2.FB # SY & SV0.FB & SV1.FB & SV2.FB # !
        SV0.FB & !SV1.FB & SV2.FB ;
    PH1 := !SV0.FB & !SV1.FB & SV2.FB ;
    PH2 := !SV0.FB & SV1.FB & !SV2.FB ;
    PH5 := SV0.FB & !SV1.FB & SV2.FB ;
    PH6 := SV0.FB & SV1.FB & !SV2.FB ;
    PH45 := SV0.FB & !SV1.FB ;
    PH123 := !SV0.FB & SV1.FB # !SV0.FB & SV2.FB ;
    PH234 := !SV0.FB & SV1.FB # SV0.FB & !SV1.FB & !SV2.FB ;
    SYNC := EN_MC & SY & !SV0.FB & !SV1.FB & !SV2.FB # EN_MC & EN_EM & !SV0.FB
        & !SV1.FB & !SV2.FB # EN_EM & SV0.FB & SV1.FB & SV2.FB # SY & SV0.FB &
        SV1.FB & SV2.FB ;
END DIV7REV1

```

### 3.6.5 VBX\_REV1

```
" D:\CDF\SRC\MASTCLK\VBX_REV1.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.02c0
" Thu Mar 13 13:22:42 1997

" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are speed optimized.

MODULE VBX_REV1

DECLARATIONS
"clock name
    CLK PIN;

"Input variables
    BXING PIN;
    ENMCLK PIN;

"Output variables
    B4V рIN ISTYPE 'reg';
    MODE132 PIN ISTYPE 'reg';
    MODE400 PIN ISTYPE 'reg';
    VBX PIN ISTYPE 'reg';

"State variables
    SV0 PIN ISTYPE 'reg';
    SV1 PIN ISTYPE 'reg';
    SV2 PIN ISTYPE 'reg';
    SV3 PIN ISTYPE 'reg';

"Clocked logic clock setup
EQUATIONS
    B4V рBX.clk=CLK;
    MODE132.clk=CLK;
    MODE400.clk=CLK;
    VBX.clk=CLK;

"State Register assignment
DECLARATIONS
    sreg=[ SV0,SV1,SV2,SV3];

EQUATIONS
    sreg.clk=CLK;

DECLARATIONS
    STATE1=[0, 0, 0, 0];
    STATE2=[0, 0, 0, 1];
    STATE3=[0, 0, 1, 0];
```

```

STATE4=[0, 0, 1, 1];
STATE5=[0, 1, 0, 0];
STATE6=[0, 1, 0, 1];
STATE7=[0, 1, 1, 0];
STATE8=[0, 1, 1, 1];
STATE9=[1, 0, 0, 0];
STATE10=[1, 0, 0, 1];
STATE12=[1, 0, 1, 0];
STATE13=[1, 0, 1, 1];
STATE14=[1, 1, 0, 0];

```

#### EQUATIONS

```

SV0 := BXING & SV0.FB & !SV1.FB & SV2.FB & !SV3.FB # ENMCLK & SV0.FB & !
SV1.FB & SV2.FB & SV3.FB # !BXING & !SV0.FB & SV1.FB & SV2.FB & SV3.FB #
!BXING & SV0.FB & !SV1.FB & !SV2.FB # BXING & !SV0.FB & !SV1.FB & SV2.FB &
SV3.FB # ENMCLK & SV0.FB & SV1.FB & !SV2.FB & !SV3.FB ;

SV1 := !BXING & !SV0.FB & !SV1.FB & !SV2.FB & SV3.FB # BXING & !SV0.FB &
SV1.FB & !SV2.FB & SV3.FB # !BXING & !SV0.FB & SV1.FB & !SV3.FB # BXING &
!SV0.FB & !SV1.FB & SV2.FB & SV3.FB # ENMCLK & SV0.FB & SV1.FB & !SV2.FB &
!SV3.FB ;

SV2 := BXING & !SV0.FB & !SV1.FB & SV2.FB & !SV3.FB # BXING & !SV0.FB & !
SV2.FB & SV3.FB # !BXING & !SV0.FB & SV1.FB & SV2.FB # !BXING & SV0.FB &
!SV1.FB & !SV2.FB & SV3.FB ;

SV3 := BXING & ENMCLK & !SV0.FB & !SV1.FB & !SV3.FB # BXING & !SV0.FB & !
SV1.FB & SV2.FB & !SV3.FB # !BXING & !SV0.FB & SV1.FB & !SV3.FB # !BXING
& SV0.FB & !SV1.FB & !SV2.FB ;

B4V р := !BXING & SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB ;

MODE132 := BXING & !SV0.FB & !SV1.FB & SV2.FB & SV3.FB # ENMCLK & SV0.FB &
SV1.FB & !SV2.FB & !SV3.FB ;

MODE400 := BXING & SV0.FB & !SV1.FB & SV2.FB & !SV3.FB # ENMCLK & SV0.FB &
!SV1.FB & SV2.FB & SV3.FB # !BXING & SV0.FB & !SV1.FB & !SV2.FB ;

VBX := !BXING & SV0.FB & !SV1.FB & !SV2.FB & SV3.FB # BXING & !SV0.FB & !
SV1.FB & SV2.FB & SV3.FB # ENMCLK & SV0.FB & SV1.FB & !SV2.FB & !SV3.FB ;
END VBX_REV1

```

## 4. RSM

### 4.1 Pin Description

Design				SRC/RSM-Readout State Machine FPGA	
Date				6-Nov-96	
				7/21/97 11:31 AM	
Signal	I	O	OT	I/O	Function
					<b>VRB Interface</b>
VRB_CMD[3:0]		4			VRB command bus
VRB_STS[9:0]	10				VRB status lines
VRB_STRB		1			VRB command bus strobe
/LATCH_VRB		1			VRB command bus data latch enable
/WRT_READ_FIFO		1			Readout, pipe, & bunch fifo write line
/RD_READ_ADD		1			Readout fifo read line
/RES_READ_FIFO		1			Readout fifo reset line
/READ_EMPTY	1				Readout fifo empty flag
/READ_FULL	1				Readout fifo full flag
/RD_PIPE		1			Pipe cap fifo read
/RD_BUNCH		1			Bunch fifo read line
<b>23</b>					
					<b>Taxi Interface</b>
/TEST	1				TSI Diagnostic sequence
/RUN	1				TSI RUN Command
/RECOVER	1				TSI RESET Command
/HALT	1				TSI HALT Command
/CONTROL	1				TSI Control word
/CAL	1				Calibration Command
CALIB[2:0]	3				Calibration Command Code
/L1A	1				L1 Accept
L1A_ADDR[1:0]	2				L1 Accept pointer
/L2A	1				L2 Accept
L2A_ADDR[1:0]	2				L2 Accept pointer
/L2R	1				L2 Reject
L2R_ADDR[1:0]	2				L2 Reject pointer
<b>18</b>					

<b>TSI Status</b>				
/L1_DONE	1			Acknowledge pipe cap return
/DONE	1			Deasserted from L2A to VRB data valid
/WAIT	1			No available SCAN buffers
/ERROR	1			Fatal error
	<b>4</b>			
<b>FIB Interface</b>				
/GLFE	1			G-Link fatal error
FIB[4:0]	5			FIB command outputs
FIB5	1			FIB Execute immediate (FXQTI)
FIB[17:16]	2			Front end chip status(RDQ code)
FIB18	1			PIPE_RD2
	<b>10</b>			
<b>Master Clock Interface</b>				
/BXING	1			Beam crossing
/BZERO	1			Bunch Zero
/GAP	1			Abort gap marker
/SYNC	1			Board level SYNC signal
/XQT	1			Board level EXECUTE signal
RF_IN	1			53 MHz CDF system clock
RSM_CLK	1			132ns symmetric clock
	<b>7</b>			
<b>VME Interface</b>				
VDAT[15:0]		16		Board level VME data bus lower byte
VADDR[4:1]	4			Board level VME address lines
/STROBE[1:0]	2			Board level VME data strobe
/BWRITE	1			Board level VME write line
/RSM	1			RSM request line from VME
/VMEOK2		1		RSM local acknowledge to VME
	<b>25</b>			
<b>Miscellaneous</b>				
/SRC_ERR[5:0]		6		Output to error handler
/SRC2FIB	1			Indicates local G-Link connection to FIB
/EN_RSM	1			Permission to use FIB output lines
/CONFIG2		1		Configuration indicator

/RES_RSM	1			Reset entire RSM FPGA
	<b>10</b>			
<b>Queue indicators</b>				
/L1_Q[3:0]		4		L1 / pipe_cap queue indicators
/L2_Q[3:0]		4		L2 / VRB READ buffer queue indicators
/L3_Q[7:0]		8		L3 / VRB SCAN queue indicators
	<b>16</b>			
<b>Buf_mng</b>				
/VRB_DAT_OE		1		VRB data output enable
/ADDR[2:1]		2		Address for VME access to pointers
/L1A_ADD[1:0]		2		L1 Accept pointer
/WRT_BP		1		Write Buffer Pointer
/L2A_ADD[1:0]		2		L2 Accept address
/L1_PFE	1			L1 Pointer fifo Empty
/L1_PFF	1			L1 Pointer fifo Full
/WRT_L1PF		1		Write L1 Pointer fifo
/RD_L1PF		1		Read L1 Pointer fifo
/RESET_BM		1		To buf_mng dedicated /RESET line
/L1_PTR[1:0]	2			L1 pointer output to RSM
/SEL_BUF		1		Select between SCAN address and EID
/P_FIFO_FULL	1			Pending fifo Full
/P_FIFO_EMPTY	1			Pending fifo Empty
/WRT_P_REG		1		Write Pending Register
/WRT_P_FIFO		1		Write Pending Fifo
/RD_P_FIFO		1		Read Pending Fifo
/P_FIFO_OE		1		Pending Fifo Output Enable
/E_FIFO_OE	1			Empty Fifo Output Enable
/E_FIFO_EMPTY	1			Empty fifo empty
/E_FIFO_FULL	1			Empty fifo full
/RD_E_FIFO		1		Read Empty fifo
/WRT_E_FIFO		1		Write Empty fifo
BM_DAT[3:0]			4	Buffer manager data io bus
/RD_BM		1		Buffer manager bus direction
/EF_DATA		1		Write Empty fifo data
/PTR_DATA		1		Write Pointer data
/PEND_2_VME		1		Read Pending fifo data to VME
/EMPTY_2_VME		1		Read Empty fifo data to VME

/PTR_2_VME		1			Read pointer data to VME
<b>37</b>	9	24	0	4	
<b>Total Pin Count</b>	<b>58</b>	<b>72</b>	<b>0</b>	<b>20</b>	<b>150</b>
<b>150</b>					

## 4.2 Constraint File

```
# ver1_2 rev1
# Nov 1, 1996
notplace instance *: U3 J16;
place instance -ADDR1_pad : R14 ;
place instance -ADDR2_pad : V17 ;
place instance -BWRITE_pad : V16 ;
place instance -CALI_pad : V11 ;
place instance -CONFIG2_pad : B2 ;
place instance -DONE_pad : F2 ;
place instance -E_FIFO_EMPTY_pad : B3 ;
place instance -E_FIFO_OE_pad : B4 ;
place instance -EF_DATA_pad : R1 ;
place instance -EMPTY_2_VME_pad : P15 ;
place instance -EN_RSM_pad : B9 ;
place instance -ERROR_pad : H18 ;
place instance -GAP_pad : K1 ;
place instance -GLFE_pad : L18 ;
# place instance -FFO_ERR0_pad : L18 ;
#place instance -FFO_ERR1_pad : R11 ;
#place instance -FFO_ERR2_pad : T18 ;
#place instance -FFO_ERR3_pad : R13 ;
place instance -HALT_pad : B18 ;
place instance -L1_DONE_pad : N17 ;
place instance -L1_PFE_pad : C10 ;
place instance -L1_PTR0_pad : C11 ;
place instance -L1_PTR1_pad : B11 ;
place instance -L1_Q0_pad : P1 ;
place instance -L1_Q1_pad : R8 ;
place instance -L1_Q2_pad : U6 ;
place instance -L1_Q3_pad : V6 ;
place instance -L1A_ADD0_pad : G18 ;
place instance -L1A_ADD1_pad : E16 ;
place instance -L1A_pad : K17 ;
place instance -L2_Q0_pad : A7 ;
place instance -L2_Q1_pad : A10 ;
place instance -L2_Q2_pad : B10 ;
place instance -L2_Q3_pad : A6 ;
place instance -L2A_ADD0_pad : F3 ;
place instance -L2A_ADD1_pad : G2 ;
place instance -L2A_pad : D5 ;
place instance -L2R_pad : E1 ;
place instance -L3_Q0_pad : M1 ;
place instance -L3_Q1_pad : M4 ;
place instance -L3_Q2_pad : N1 ;
place instance -L3_Q3_pad : N2 ;
place instance -L3_Q4_pad : G1 ;
place instance -L3_Q5_pad : H3 ;
place instance -L3_Q6_pad : K3 ;
place instance -L3_Q7_pad : L1 ;
place instance -LATCH_VRB_pad : T8 ;
place instance -P_FIFO_EMPTY_pad : D8 ;
```

```

place instance -P_FIFO_OE_pad      : T9 ;
place instance -PEND_2_VME_pad     : U14 ;
place instance -PTR_2_VME_pad     : N15 ;
place instance -PTR_DATA_pad      : U4 ;
place instance -RD_BM_pad         : N16 ;
place instance -RD_BUNCH_pad     : R17 ;
place instance -RD_E_FIFO_pad    : C5 ;
place instance -RD_L1PF_pad       : A11 ;
place instance -RD_P_FIFO_pad    : V7 ;
place instance -RD_PIPE_pad       : N18 ;
place instance -RD_READ_ADD_pad  : R12 ;
place instance -RECOVER_pad       : D17 ;
place instance -RES_READ_FIFO_pad : M2 ;
place instance -RES_RSM_pad       : A3 ;
place instance -RESET_BM          : T11 ;
#place instance -OLD_RESET_pad   : B7 ;
place instance -RSM_pad           : U9 ;
place instance -RUN_pad           : E17 ;
place instance -SEL_BUF_pad       : V9 ;
place instance -SRC2FIB_pad       : T6 ;
place instance -SRC_ERR0_pad      : J1 ;
place instance -SRC_ERR1_pad      : M17 ;
place instance -SRC_ERR2_pad      : M18 ;
place instance -SRC_ERR3_pad      : M15 ;
place instance -SRC_ERR4_pad      : G17 ;
place instance -SRC_ERR5_pad      : B1 ;
place instance -STROBE0_pad       : E3 ;
place instance -STROBE1_pad       : C17 ;
place instance -VMEOK2_pad        : K18 ;
place instance -VRB_DAT_OE_pad   : T10 ;
place instance -WAIT_pad          : C6 ;
place instance -WRT_BP_pad        : T5 ;
place instance -WRT_E_FIFO_pad   : P3 ;
place instance -WRT_L1PF_pad      : H16 ;
place instance -WRT_P_FIFO_pad   : J3 ;
place instance -WRT_P_REG_pad    : A2 ;
place instance -WRT_READ_FIFO_pad: H17 ;
place instance -XQT_pad           : V2 ;
place instance BM_DAT0_pad        : T15 ;
place instance BM_DAT1_pad        : U16 ;
place instance BM_DAT2_pad        : T14 ;
place instance BM_DAT3_pad        : U15 ;
place instance -BXING_pad         : H1 ;
place instance FIB0_pad           : B8 ;
place instance FIB1_pad           : B5 ;
place instance FIB2_pad           : A4 ;
place instance FIB3_pad           : V8 ;
place instance FIB4_pad           : A8 ;
place instance FIB5_pad           : D7 ;
place instance FIB16_pad          : C8 ;
place instance FIB17_pad          : L4 ;
place instance FIB18_pad          : U8 ;
place instance L1A_ADDR0_pad      : F18 ;
place instance L1A_ADDR1_pad      : F16 ;
place instance L2A_ADDR0_pad      : F4 ;

```

```

place instance L2A_ADDR1_pad      : H4 ;
place instance L2R_ADDR0_pad      : D1 ;
place instance L2R_ADDR1_pad      : C1 ;
place instance QDR0_pad          : U12 ;
place instance QRD1_pad          : U11 ;
place instance RF_IN_pad          : C3 ;
place instance RSM_CLK_pad       : B17 ;
place instance SYNC_pad          : K2 ;
place instance VADDR1_pad        : R18 ;
place instance VADDR2_pad        : T13 ;
place instance VADDR3_pad        : V14 ;
place instance VADDR4_pad        : V15 ;
place instance VADDR5_pad        : P17 ;
place instance VDAT0_pad         : B16 ;
place instance VDAT1_pad         : A17 ;
place instance VDAT2_pad         : C14 ;
place instance VDAT3_pad         : B15 ;
place instance VDAT4_pad         : A16 ;
place instance VDAT5_pad         : B14 ;
place instance VDAT6_pad         : C13 ;
place instance VDAT7_pad         : A15 ;
place instance VDAT8_pad         : C2 ;
place instance VDAT9_pad         : A14 ;
place instance VDAT10_pad        : B13 ;
place instance VDAT11_pad        : D14 ;
place instance VDAT12_pad        : D13 ;
place instance VDAT13_pad        : A13 ;
place instance VDAT14_pad        : B12 ;
place instance VDAT15_pad        : A12 ;
place instance VRB_CMD0_pad      : P18 ;
place instance VRB_CMD1_pad      : U10 ;
place instance VRB_CMD2_pad      : V10 ;
place instance VRB_CMD3_pad      : C4 ;
place instance VRB_STRB_pad      : L2 ;
place instance VRB_STS0_pad       : A9 ;
place instance VRB_STS1_pad       : C9 ;
place instance VRB_STS2_pad       : L16 ;
place instance VRB_STS3_pad       : L17 ;
place instance VRB_STS4_pad       : L15 ;
place instance VRB_STS5_pad       : K16 ;
place instance VRB_STS6_pad       : H15 ;
place instance VRB_STS7_pad       : G15 ;
place instance VRB_STS8_pad       : J17 ;
place instance VRB_STS9_pad       : J18 ;

```

#### ***4.3 Placement Report***

Error! Not a valid filename.

#### ***4.4 XACT Performance***

Error! Not a valid filename.

## 4.5 State CAD Diagrams

### 4.5.1 VME\_INT

```

\cdfsrc\rsm\vme_int.dia
VME Interface
10/9/96 JO
REVISED 8/1/97 NF
(view:70%)

%R0% = ADDR[^\h0]                                ADDR[] = 5:0
                                                CONTROL register: MODE, SUB-MODE (R/W)

%R2% = ADDR[^\h2] & !WRITE                         STATUS Register (READ Only)

%R4% = ADDR[^\h4]                                  Fatal Error Mask (R/W)

%R6% = ADDR[^\h6] & (!WRITE # VME_ACCESS)          VRB Scan Buffer Count (R/W)

%R8% = ADDR[^\h8] & (!WRITE # VME_ACCESS)          SVXIII History Depth (R/W)

%RA% = ADDR[^\hA] & (!WRITE # VME_ACCESS)          SVXIII Digitize Timer/Counter (R/W)

%R10% = ADDR[^\h10] & !WRITE                        Buffer Pointer 0 (READ Only)

%R12% = ADDR[^\h12] & !WRITE                        Buffer Pointer 1 (READ Only)

%R14% = ADDR[^\h14] & !WRITE                        Buffer Pointer 2 (READ Only)

%R16% = ADDR[^\h16] & !WRITE                        Buffer Pointer 3 (READ Only)

%R18% = ADDR[^\h18] & !WRITE                        EMPTY Fifo (READ Only)

%R1A% = ADDR[^\h1A] & !WRITE                        PENDING FIFO (READ Only)

%R20% = ADDR[^\h20] & (!WRITE # VME_ACCESS)          Preamp RESET Timer/Counter (R/W)

%R22% = ADDR[^\h22] & (!WRITE # VME_ACCESS)          SVXIII Reset Timer/Counter (R/W)

%R24% = ADDR[^\h24] & (!WRITE # VME_ACCESS)          Portcard Reset Timer/Counter (R/W)

%R26% = ADDR[^\h26] & (!WRITE # VME_ACCESS)          FIB Reset Timer/Counter (R/W)

%R28% = ADDR[^\h28] & !WRITE                        RSM State Machine State Number (READ ONLY)

%R30% = ADDR[^\h30] & !WRITE                        L1A L2R READOUT COMPLETE STATUS
                                                    (READ ONLY)

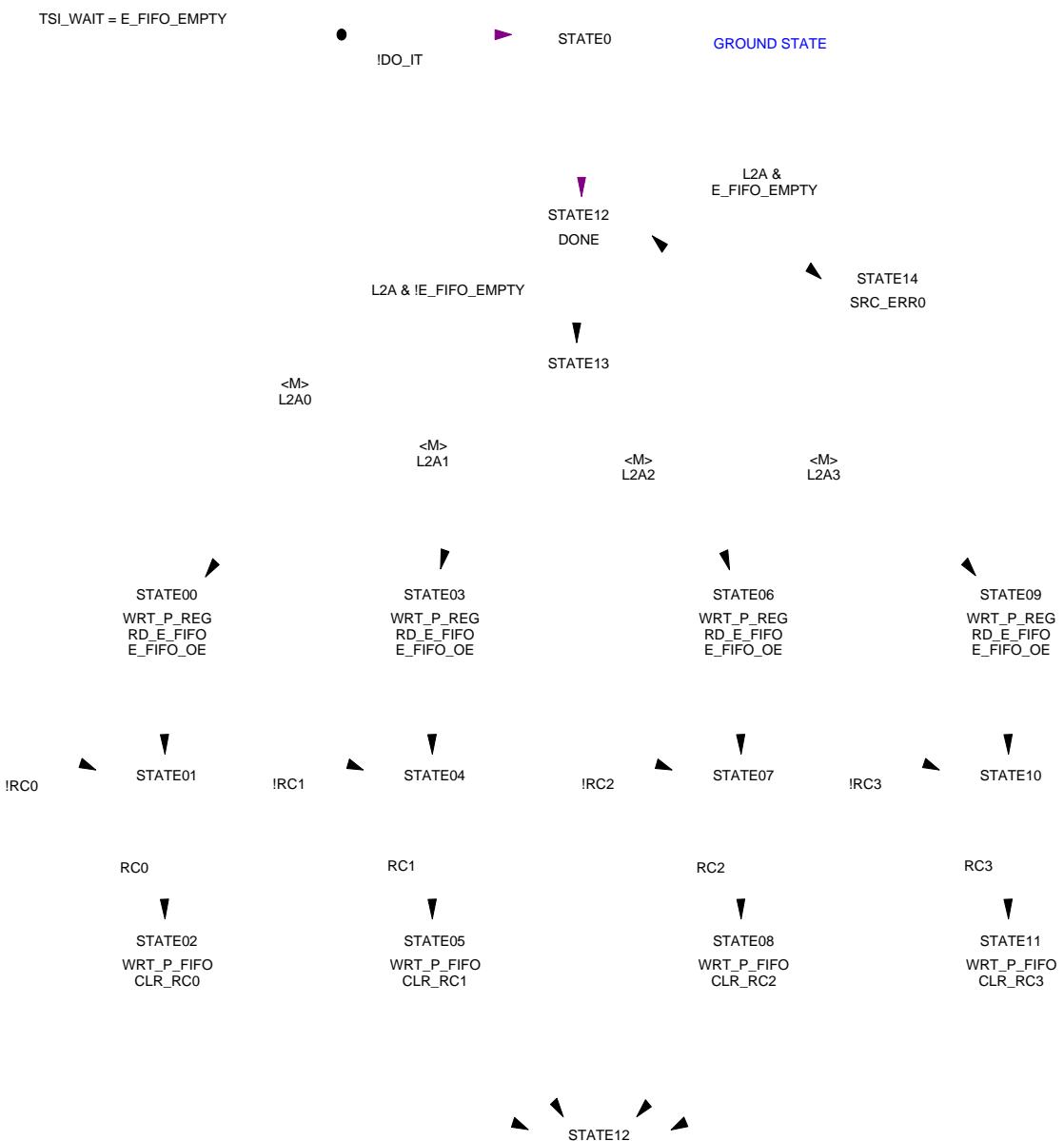
%R32% = ADDR[^\h32] & WRITE                          CAL INJECT

                                                REG0 = R0 & REQ      REG4 = R4 & REQ      REG8 = R8 & REQ      REG20 = R20 & REQ     REG24 = R24 & REQ      REG28 = R28 & REQ
                                                REG2 = R2 & REQ      REG6 = R6 & REQ      REGA = RA & REQ      REG22 = R22 & REQ     REG26 = R26 & REQ      REG30 = R30 & REQ
                                                BUFLMAN = REQ&(R10 or R12 or R14 or R16 or R18 or R1A)
                                                PEND2VME = REQ & R1A
                                                EMPTY2VME = REQ & R18
                                                PTR2VME = REQ & (R10 or R12 or R14 or R16)
                                                VMEOK = REQ & (R0 or R2 or R4 or R6 or R8 or RA or R10 or R12 or R14 or R16 or R18 or R1A or R28 or R30 or R20 or R22 or R24 or R26 or R32)
                                                DATA_OUT = VMEOK & !WRITE
                                                DATA_IN = VMEOK & WRITE

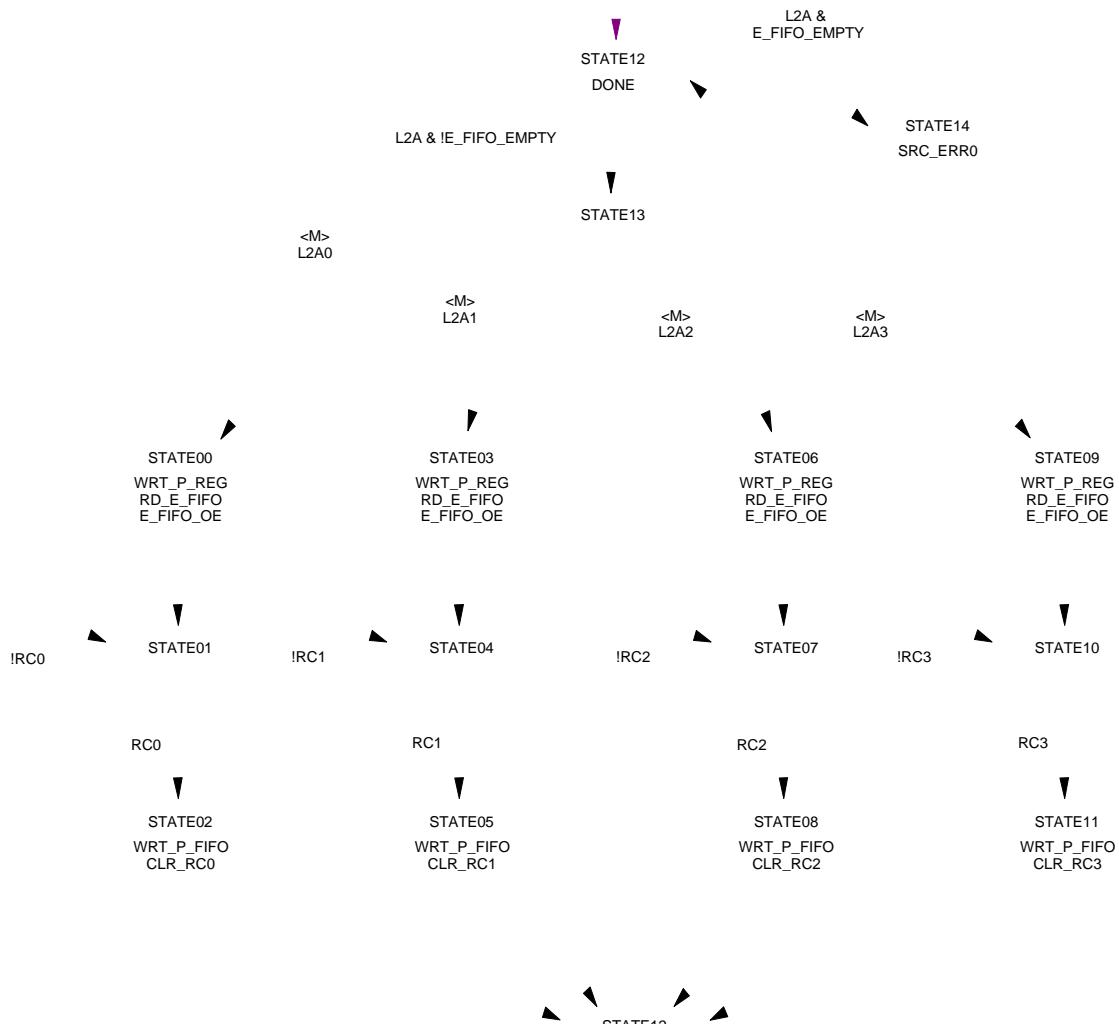
```

#### 4.5.2 L2A\_SM

\cdf\src\rsm\L2A\_SM.dia  
L2 Accept State Machine  
Oct. 10, 1996 JO/CG  
(view:70% portrait)



\cdflsrc\rsm\L2A\_SM.dia  
L2 Accept State Machine  
Oct. 10, 1996 JO/CG  
(view:70% portrait)

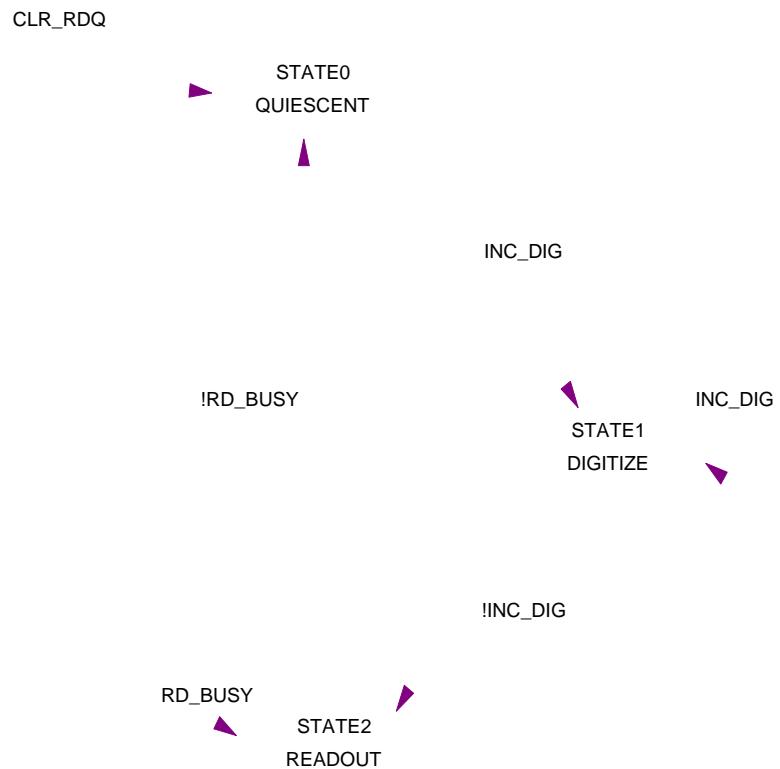


#### 4.5.3 RDQ\_SM

\cdf\src\rsm\rdq\_sm.dia  
Back end status state machine  
Oct. 8, 1996 J.Oliver  
(use 70% portrait)

RDQ[] = 1:0

```
%QUIESCENT% = RDQ[0]  
%DIGITIZE% = RDQ[1]  
%READOUT% = RDQ[2]
```





#### 4.5.4 READ\_Q

\cdfsrc\rsm\read\_q.dia  
 Nov.5, 1996 JO  
 Read queue or L2 queue thermometer  
 indicators.  
 Revised 8/1/97 NF

L2Q[] = 3:0

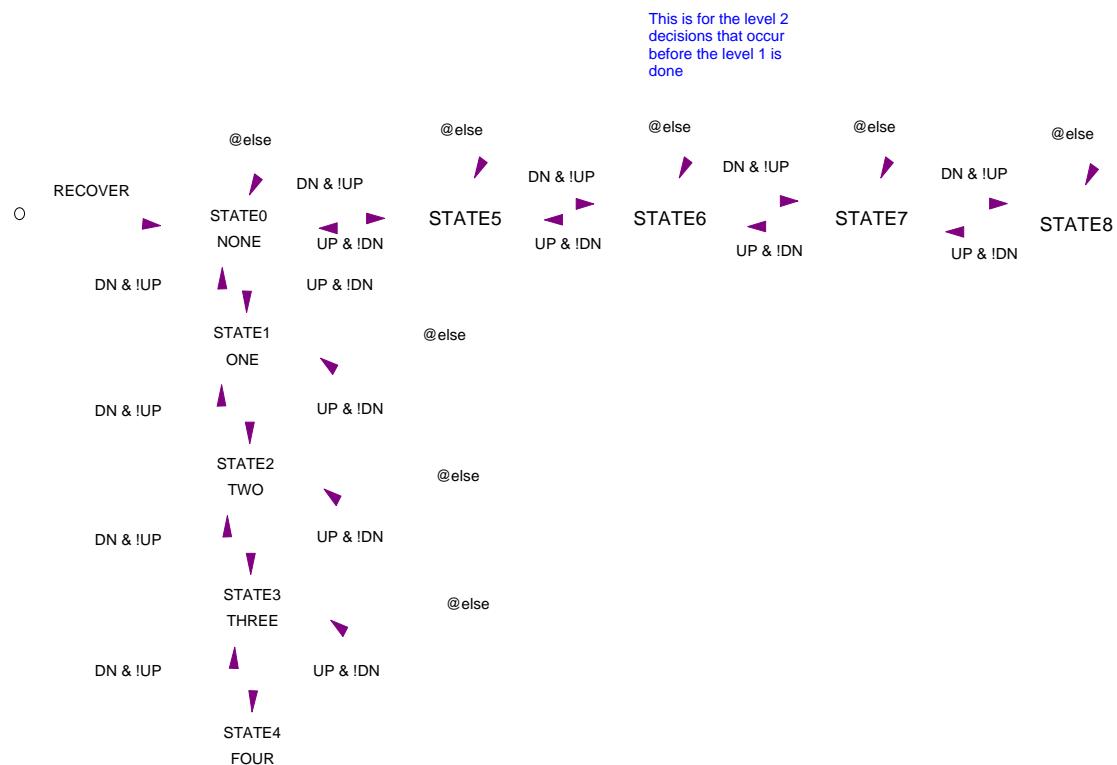
%NONE% = L2Q[^hF]

%ONE% = L2Q[7]

%TWO% = L2Q[3]

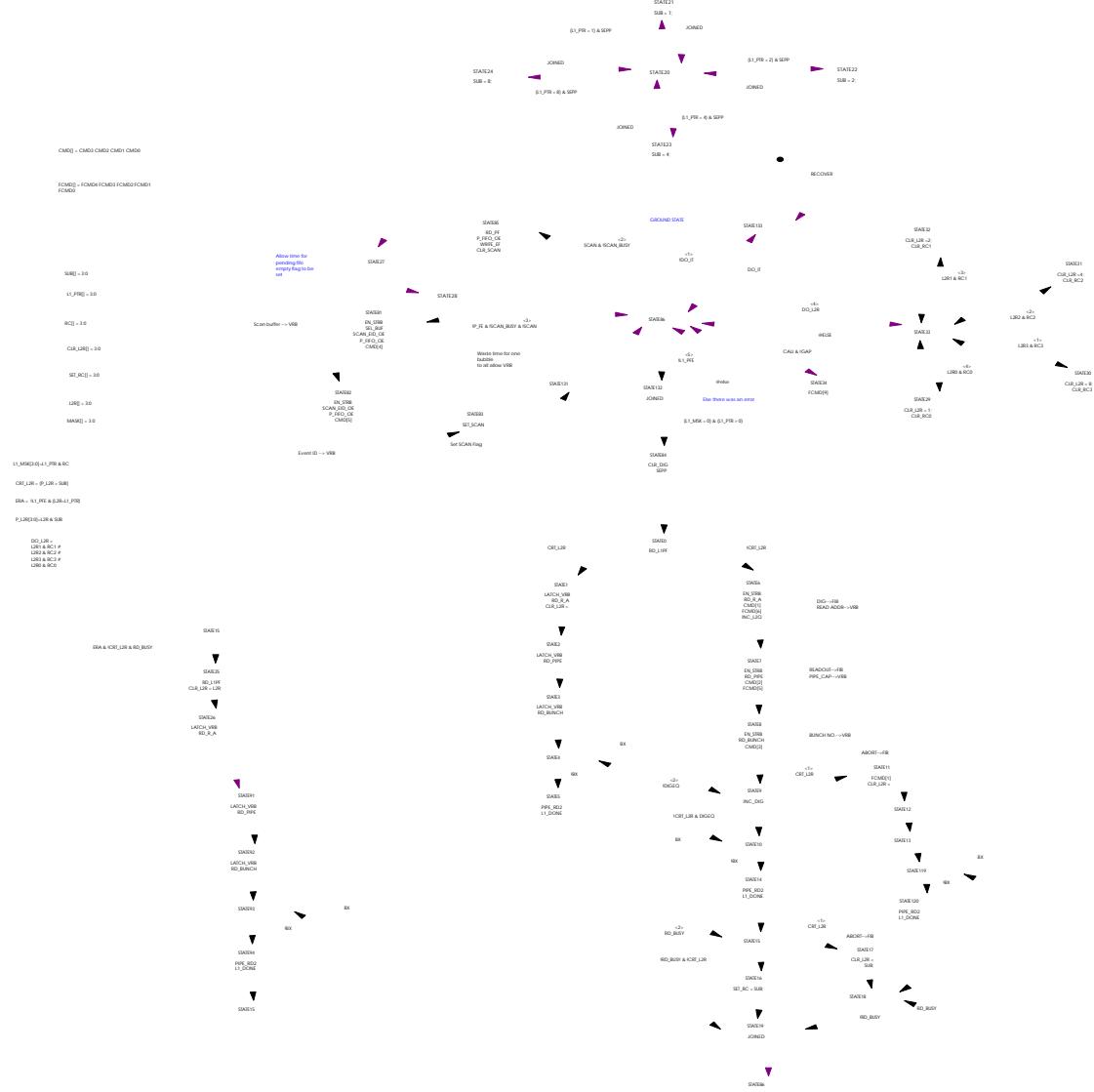
%THREE% = L2Q[1]

%FOUR% = L2Q[0]



#### 4.5.5 READ\_SM6

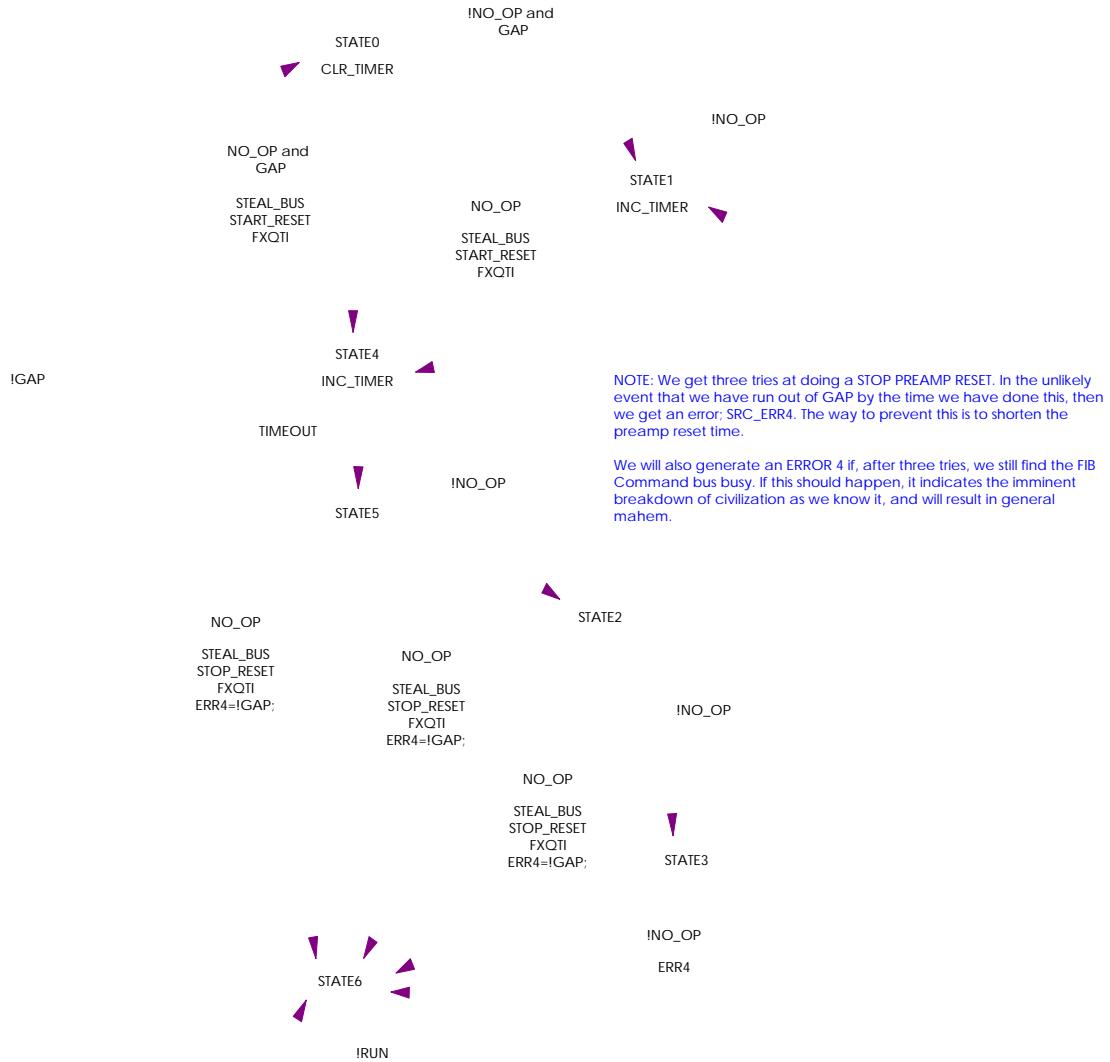
READ\_SM  
Readout State Machine  
June 6, 1996  
J. Oliver, M. Spiropulu  
Revised 8/1/97 NF



#### 4.5.6 RES\_PR

```
\ncdf\src\vsm\res_pr.dia
Sends START PREAMP RESET FCMD[4] and STOP PREAMP RESET FCMD[2]
at the appropriate place in the gap if, and only if, the FIB Command bus is not
already in use by the READ_SM.
Up to three attempts are made to do a START PR. If the bus is busy all three times,
we give up and try again later.
Oct. 25, 1996
```

```
FCMD[] = 4:0
%STOP_RESET% = FCMD[2]
%START_RESET% = FCMD[4]
```



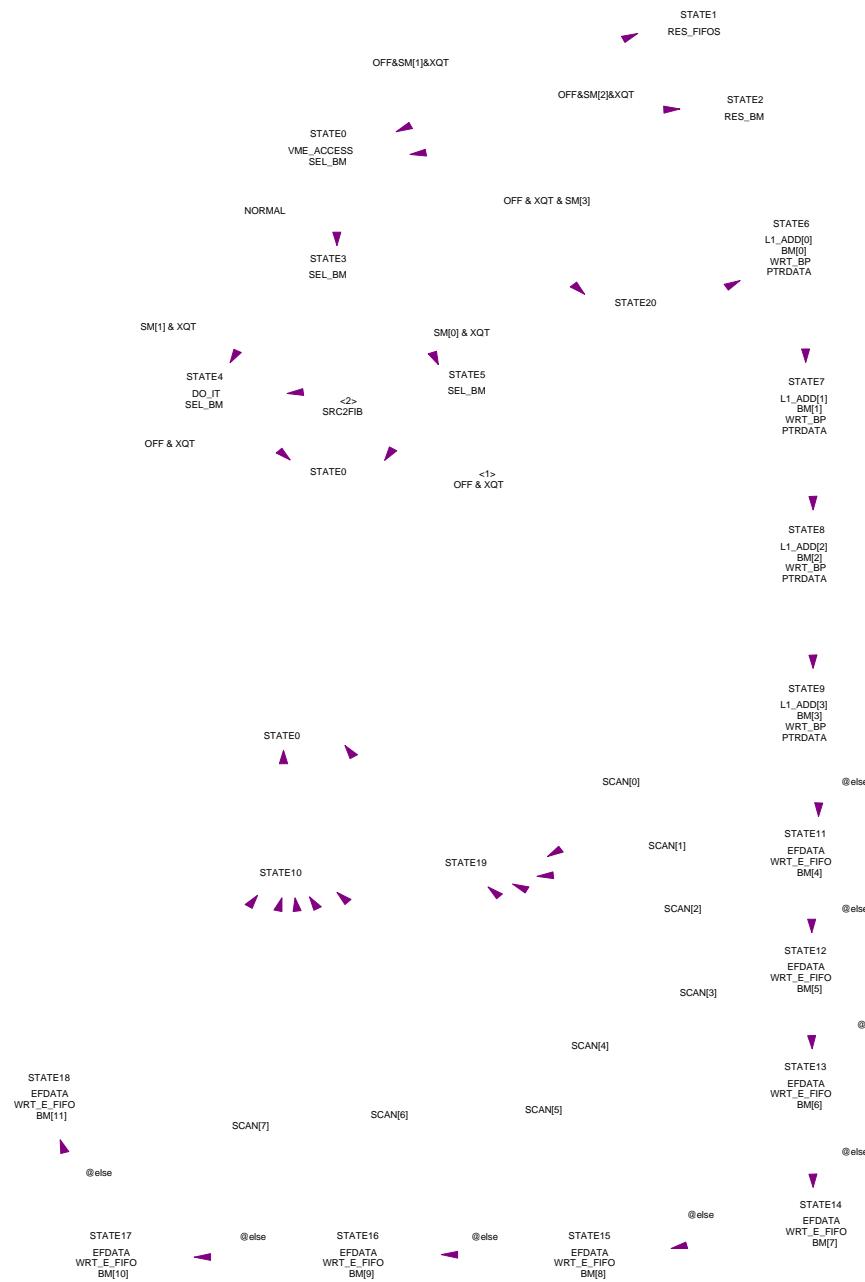
#### 4.5.7 RSMMODES

\cdf\src\rsm\rsmmodes.dia  
Oct. 11, 1996  
J.Oliver  
Revised 4/11/97

```

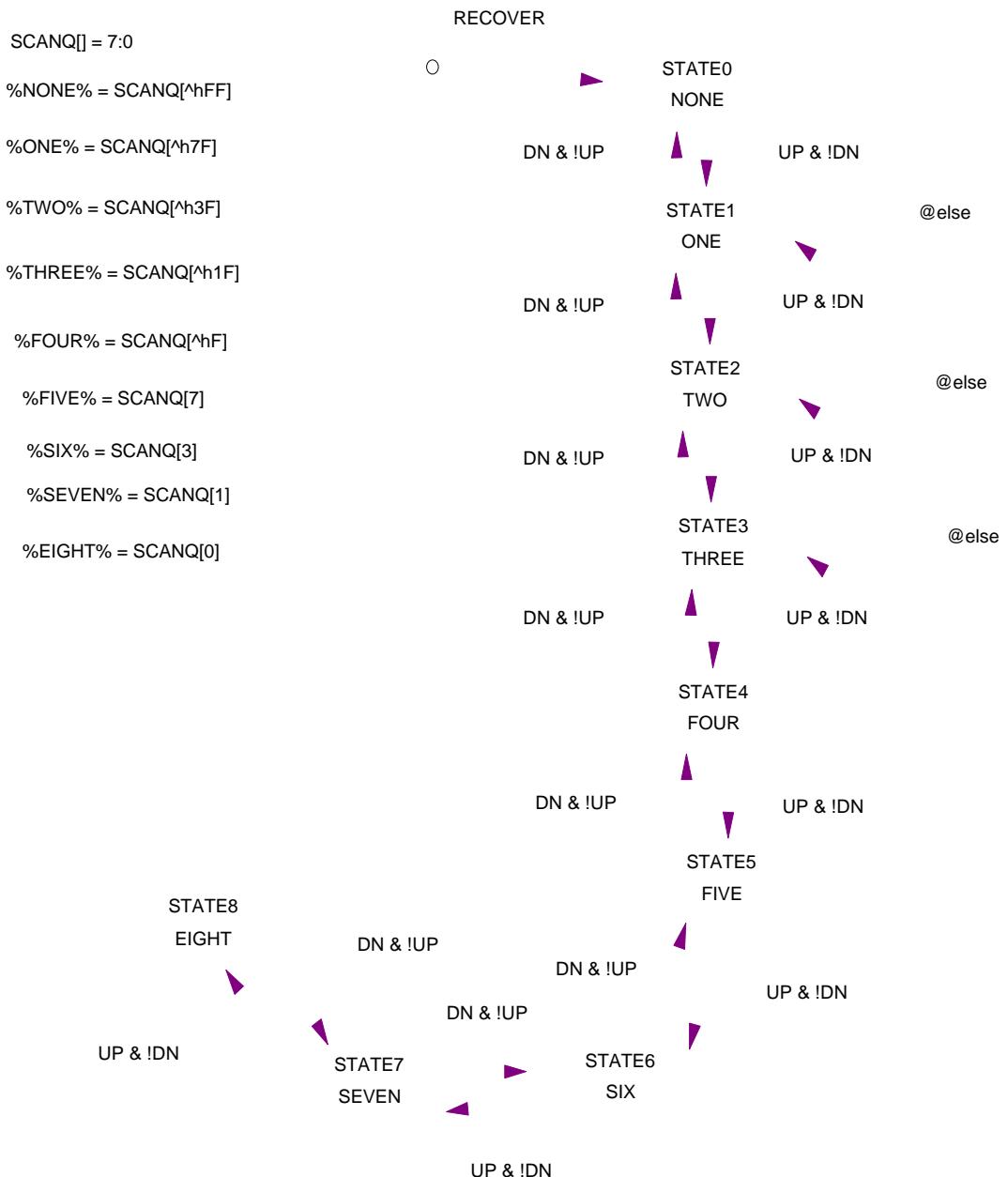
MODE[] = CREG7 CREG6 CREG5 CREG4
SM[] = CREG3 CREG2 CREG1 CREG0
L1_ADD[] = 1:0
BM[] = 3:0
SCAN[] = 3:0
%OFF% = MODE[0]
%NORMAL% = MODE[3]

```



#### 4.5.8 SCAN Q

\cdf\src\rsm\scan\_q.dia  
Nov.5, 1996 JO  
SCAN queue thermometer indicators.

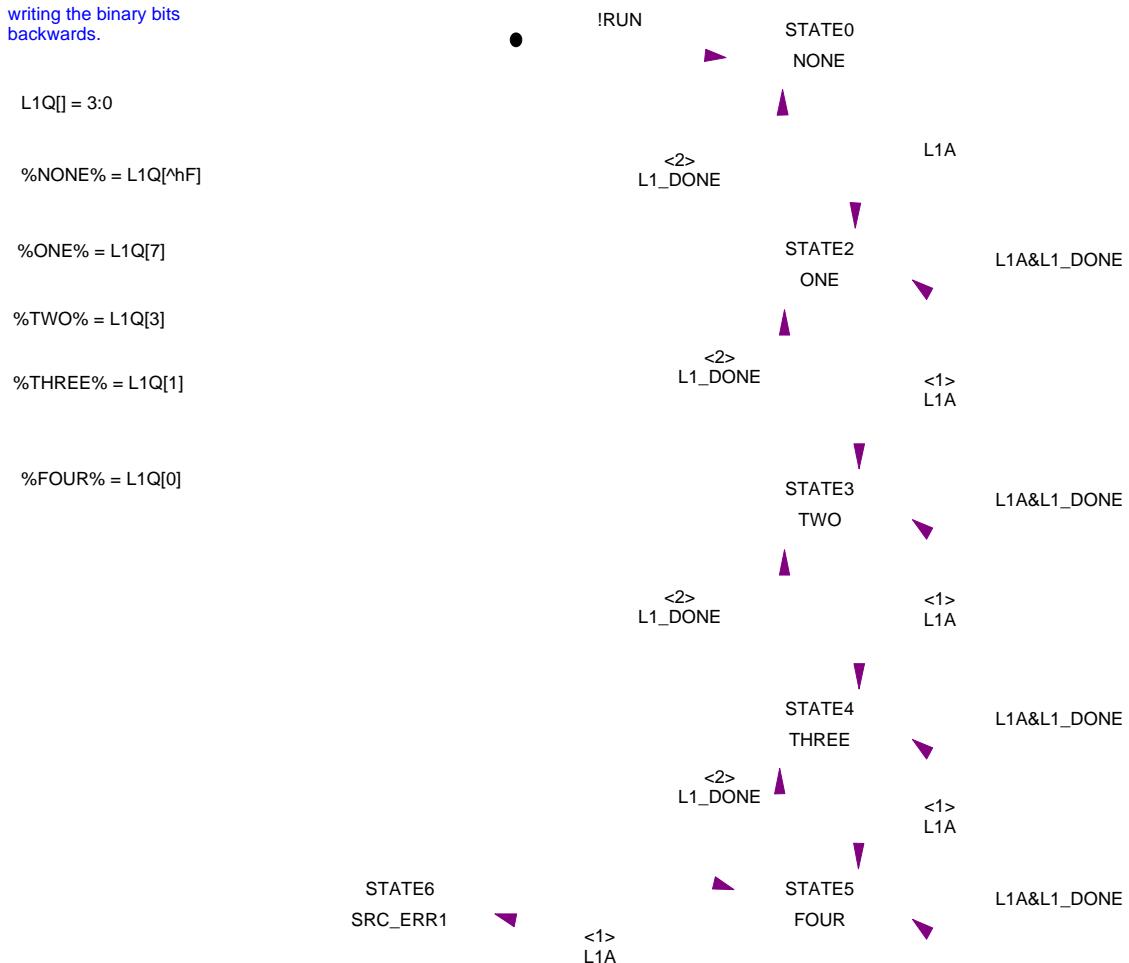


#### 4.5.9 SRC\_ERR1

\cdf\src\rsm\src\_err1.dia  
Oct. 10, 1996 JO

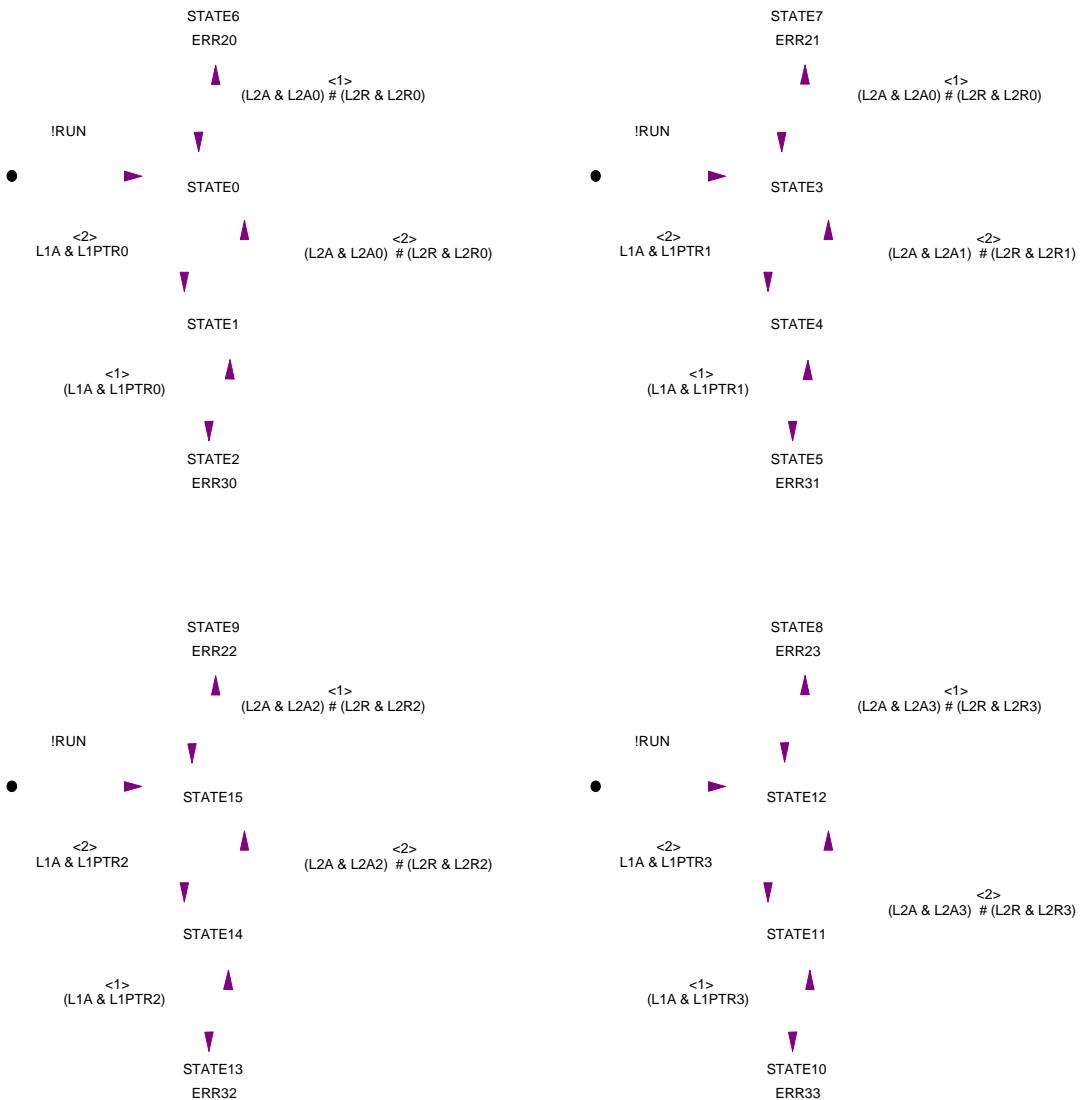
Generates an error on receiving L1A when 4 L1 cells are already active.  
Don't try this at home!

Thermometer code is obtained by complementing the desired number and writing the binary bits backwards.



#### 4.5.10 SRC\_ERR2

```
\cdflsrc\lsm\src_err2.dia
Oct. 10, 1996
Generates 2 kinds of errors
ERR2: L2 decision on buffer not yet accepted for L1
ERR3: L1 decision on buffer not vacated by L2 decision
```



## 4.6 HDL Code

### 4.6.1 VME\_INT

```
" D:\CDF\SRC\RSM\VME_INT.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.2
" Thu Nov 20 10:14:34 1997

" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are area optimized.
```

```
MODULE VME_INT
```

```
DECLARATIONS
```

```
"Input variables
```

```
    ADDR0 PIN;
    ADDR1 PIN;
    ADDR2 PIN;
    ADDR3 PIN;
    ADDR4 PIN;
    ADDR5 PIN;
    REQ PIN;
    VME_ACCESS PIN;
    WRITE PIN;
```

```
"Logic variables
```

```
    BUFLIN PIN ISTYPE 'com';
    DATA_IN PIN ISTYPE 'com';
    DATA_OUT PIN ISTYPE 'com';
    EMPTY2VME PIN ISTYPE 'com';
    PEND2VME PIN ISTYPE 'com';
    PTR2VME PIN ISTYPE 'com';
    REG0 PIN ISTYPE 'com';
    REG2 PIN ISTYPE 'com';
    REG4 PIN ISTYPE 'com';
    REG6 PIN ISTYPE 'com';
    REG8 PIN ISTYPE 'com';
    REG20 PIN ISTYPE 'com';
    REG22 PIN ISTYPE 'com';
    REG24 PIN ISTYPE 'com';
    REG26 PIN ISTYPE 'com';
    REG28 PIN ISTYPE 'com';
    REG30 PIN ISTYPE 'com';
    REG32 PIN ISTYPE 'com';
    REGA PIN ISTYPE 'com';
    VMEOK PIN ISTYPE 'com';
```

```
"Vectors
```

```

DECLARATIONS
ADDR=[  

    ADDR5,  

    ADDR4,  

    ADDR3,  

    ADDR2,  

    ADDR1,  

    ADDR0  

];
"Logic Equations
EQUATIONS
BUFMAN = !WRITE & !ADDR5 & ADDR4 & !ADDR3 & !ADDR0 & REQ # !WRITE & !ADDR5  

& ADDR4 & !ADDR2 & !ADDR0 & REQ ;  

DATA_IN = VMEOK & WRITE ;  

DATA_OUT = VMEOK & !WRITE ;  

EMPTY2VME = !WRITE & !ADDR5 & ADDR4 & ADDR3 & !ADDR2 & !ADDR1 & !ADDR0 & REQ  

;  

PEND2VME = !WRITE & !ADDR5 & ADDR4 & ADDR3 & !ADDR2 & ADDR1 & !ADDR0 & REQ ;  

PTR2VME = !WRITE & !ADDR5 & ADDR4 & !ADDR3 & !ADDR0 & REQ ;  

REG0 = REQ & !ADDR5 & !ADDR4 & !ADDR3 & !ADDR2 & !ADDR1 & !ADDR0 ;  

REG2 = !WRITE & REQ & !ADDR5 & !ADDR4 & !ADDR3 & !ADDR2 & ADDR1 & !ADDR0 ;  

REG4 = REQ & !ADDR5 & !ADDR4 & !ADDR3 & ADDR2 & !ADDR1 & !ADDR0 ;  

REG6 = REQ & !WRITE & !ADDR5 & !ADDR4 & !ADDR3 & ADDR2 & ADDR1 & !ADDR0 #  

REQ & VME_ACCESS & !ADDR5 & !ADDR4 & !ADDR3 & ADDR2 & ADDR1 & !ADDR0 ;  

REG8 = REQ & !WRITE & !ADDR5 & !ADDR4 & ADDR3 & !ADDR2 & !ADDR1 & !ADDR0 #  

REQ & VME_ACCESS & !ADDR5 & !ADDR4 & ADDR3 & !ADDR2 & !ADDR1 & !ADDR0  

;  

REG20 = REQ & !WRITE & ADDR5 & !ADDR4 & !ADDR3 & !ADDR2 & !ADDR1 & !ADDR0 #  

REQ & VME_ACCESS & ADDR5 & !ADDR4 & !ADDR3 & !ADDR2 & !ADDR1 & !ADDR0  

;  

REG22 = REQ & !WRITE & ADDR5 & !ADDR4 & !ADDR3 & !ADDR2 & ADDR1 & !ADDR0 #  

REQ & VME_ACCESS & ADDR5 & !ADDR4 & !ADDR3 & !ADDR2 & ADDR1 & !ADDR0 ;  

REG24 = REQ & !WRITE & ADDR5 & !ADDR4 & !ADDR3 & ADDR2 & !ADDR1 & !ADDR0 #  

REQ & VME_ACCESS & ADDR5 & !ADDR4 & !ADDR3 & ADDR2 & !ADDR1 & !ADDR0 ;  

REG26 = REQ & !WRITE & ADDR5 & !ADDR4 & !ADDR3 & ADDR2 & ADDR1 & !ADDR0 #  

REQ & VME_ACCESS & ADDR5 & !ADDR4 & !ADDR3 & ADDR2 & ADDR1 & !ADDR0 ;  

REG28 = !WRITE & REQ & ADDR5 & !ADDR4 & ADDR3 & !ADDR2 & !ADDR1 & !ADDR0 ;  

REG30 = !WRITE & REQ & ADDR5 & ADDR4 & !ADDR3 & !ADDR2 & !ADDR1 & !ADDR0 ;

```

```

REG32 = WRITE & REQ & ADDR5 & ADDR4 & !ADDR3 & !ADDR2 & ADDR1 & !ADDR0 ;

REGA = REQ & !WRITE & !ADDR5 & !ADDR4 & ADDR3 & !ADDR2 & ADDR1 & !ADDR0 #
      REQ & VME_ACCESS & !ADDR5 & !ADDR4 & ADDR3 & !ADDR2 & ADDR1 & !ADDR0 ;

VMEOK = WRITE & ADDR5 & ADDR4 & !ADDR3 & !ADDR2 & ADDR1 & !ADDR0 & REQ # !
      WRITE & ADDR5 & !ADDR4 & !ADDR3 & !ADDR0 & REQ # VME_ACCESS & ADDR5 &
!
      ADDR4 & !ADDR3 & !ADDR0 & REQ # !WRITE & ADDR5 & !ADDR3 & !ADDR2 &
!ADDR1 &
      !ADDR0 & REQ # !WRITE & ADDR5 & !ADDR4 & !ADDR2 & !ADDR1 & !ADDR0 &
REQ #
      !WRITE & !ADDR5 & ADDR4 & ADDR3 & !ADDR2 & !ADDR0 & REQ # !WRITE &
!ADDR5
      & ADDR4 & !ADDR3 & ADDR2 & !ADDR0 & REQ # !WRITE & !ADDR5 & ADDR4 &
!ADDR2
      & ADDR1 & !ADDR0 & REQ # !WRITE & ADDR4 & !ADDR3 & !ADDR2 & !ADDR1 & !
ADDR0 & REQ # !WRITE & !ADDR5 & ADDR3 & !ADDR2 & ADDR1 & !ADDR0 & REQ
# !
      WRITE & !ADDR4 & ADDR3 & !ADDR2 & !ADDR1 & !ADDR0 & REQ # VME_ACCESS
& !
      ADDR5 & !ADDR4 & ADDR3 & !ADDR2 & !ADDR0 & REQ # VME_ACCESS & !ADDR4
& !
      ADDR3 & ADDR2 & ADDR1 & !ADDR0 & REQ # !WRITE & !ADDR4 & !ADDR3 &
ADDR1 & !
      ADDR0 & REQ # !ADDR5 & !ADDR4 & !ADDR3 & !ADDR1 & !ADDR0 & REQ ;

```

END VME\_INT

#### 4.6.2 L2A\_SM

```
" D:\CDF\SRC\RSM\L2A_SM.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.2
" Mon Nov 17 16:32:19 1997
```

```
" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are area optimized.
```

```
MODULE L2A_SM
TITLE 'L2A'
```

##### DECLARATIONS

```
"clock name
CLK PIN;
```

##### "Input variables

```
DO_IT PIN;
DUMP_Q PIN;
E_FIFO_EMPTY PIN;
L2A PIN;
L2A0 PIN;
L2A1 PIN;
L2A2 PIN;
L2A3 PIN;
RC0 PIN;
RC1 PIN;
RC2 PIN;
RC3 PIN;
```

##### "Output variables

```
CLR_RC0 PIN ISTYPE 'com';
CLR_RC1 PIN ISTYPE 'com';
CLR_RC2 PIN ISTYPE 'com';
CLR_RC3 PIN ISTYPE 'com';
DONE PIN ISTYPE 'com';
E_FIFO_OE PIN ISTYPE 'com';
RD_E_FIFO PIN ISTYPE 'com';
SRC_ERR0 PIN ISTYPE 'com';
WRT_P_FIFO PIN ISTYPE 'com';
WRT_P_REG PIN ISTYPE 'com';
```

##### "Logic variables

```
TSI_WAIT PIN ISTYPE 'com';
```

##### "State variables

```
SV0 PIN ISTYPE 'reg';
SV1 PIN ISTYPE 'reg';
SV2 PIN ISTYPE 'reg';
SV3 PIN ISTYPE 'reg';
```

```

"State Register assignment
DECLARATIONS
    sreg=[ SV0,SV1,SV2,SV3];

EQUATIONS
    sreg.clk=CLK;

DECLARATIONS
    STATE0=[0, 0, 0, 0];
    STATE00=[0, 0, 0, 1];
    STATE01=[0, 0, 1, 0];
    STATE02=[0, 0, 1, 1];
    STATE03=[0, 1, 0, 0];
    STATE04=[0, 1, 0, 1];
    STATE05=[0, 1, 1, 0];
    STATE06=[0, 1, 1, 1];
    STATE07=[1, 0, 0, 0];
    STATE08=[1, 0, 0, 1];
    STATE09=[1, 0, 1, 0];
    STATE10=[1, 0, 1, 1];
    STATE11=[1, 1, 0, 0];
    STATE12=[1, 1, 0, 1];
    STATE13=[1, 1, 1, 0];
    STATE14=[1, 1, 1, 1];

EQUATIONS
    SV0 := !DUMP_Q & L2A3 & SV0.FB & !SV3.FB # !DUMP_Q & !DO_IT & !SV1.FB & !
        SV2.FB & !SV3.FB # !DUMP_Q & !SV0.FB & SV2.FB & SV3.FB # !DUMP_Q & !
        SV0.FB & SV1.FB & SV2.FB # !DUMP_Q & SV0.FB & !SV1.FB # !DUMP_Q & SV1.FB
        & SV2.FB & SV3.FB # !L2A1 & !L2A0 & !L2A2 & !DUMP_Q & SV0.FB & !SV3.FB #
        !DUMP_Q & SV0.FB & !SV2.FB ;

    SV1 := !DUMP_Q & L2A1 & SV1.FB & !SV3.FB # !DUMP_Q & L2A2 & SV1.FB & !
        SV3.FB # !DUMP_Q & RC3 & !SV1.FB & SV2.FB & SV3.FB # !DUMP_Q & !DO_IT & !
        SV0.FB & !SV2.FB & !SV3.FB # !DUMP_Q & !SV0.FB & !SV1.FB & SV2.FB & SV3.FB
        # !DUMP_Q & !SV0.FB & SV1.FB & !SV3.FB # !DUMP_Q & SV0.FB & !SV2.FB &
        SV3.FB # !DUMP_Q & SV0.FB & SV1.FB & SV3.FB # !L2A3 & !L2A0 & !DUMP_Q &
        SV1.FB & !SV3.FB # !DUMP_Q & SV1.FB & !SV2.FB ;

    SV2 := !DUMP_Q & !SV0.FB & !SV1.FB & !SV2.FB & SV3.FB # !DUMP_Q & RC1 & !
        SV0.FB & !SV2.FB & SV3.FB # !DUMP_Q & L2A2 & SV0.FB & SV2.FB & !SV3.FB #
        !DUMP_Q & L2A3 & SV0.FB & SV2.FB & !SV3.FB # !DUMP_Q & !SV1.FB & SV2.FB & !
        SV3.FB # !DUMP_Q & !RC3 & SV0.FB & !SV1.FB & SV2.FB # !L2A1 & !L2A0 & !
        DUMP_Q & SV0.FB & SV2.FB & !SV3.FB # !DUMP_Q & L2A & SV0.FB & SV1.FB & !
        SV2.FB & SV3.FB ;

    SV3 := !DUMP_Q & L2A0 & SV0.FB & SV2.FB & !SV3.FB # !DUMP_Q & RC0 & !SV1.FB
        & SV2.FB & !SV3.FB # !DUMP_Q & !RC1 & !SV0.FB & SV1.FB & !SV2.FB # !
        DUMP_Q & L2A2 & SV0.FB & SV2.FB & !SV3.FB # !DUMP_Q & RC2 & SV0.FB & !
        SV1.FB & !SV3.FB # !DUMP_Q & SV0.FB & !SV1.FB & SV2.FB & !SV3.FB # !
        DUMP_Q & !RC3 & SV0.FB & !SV1.FB & SV2.FB # !DUMP_Q & !DO_IT & !SV0.FB & !
        SV2.FB & !SV3.FB # !DUMP_Q & !SV0.FB & !SV1.FB & SV2.FB & SV3.FB # !
        DUMP_Q & !SV0.FB & SV1.FB & !SV3.FB # !DUMP_Q & SV0.FB & !SV1.FB & !SV2.FB

```

```

& SV3.FB # !DUMP_Q & SV1.FB & !SV2.FB & !SV3.FB # !L2A & !DUMP_Q & SV0.FB
& !SV2.FB & SV3.FB # !DUMP_Q & SV0.FB & SV1.FB & SV2.FB & SV3.FB # !
DUMP_Q & E_FIFO_EMPTY & SV0.FB & !SV2.FB & SV3.FB ;

CLR_RC0= SV3.FB & SV2.FB & !SV1.FB & !SV0.FB ;
CLR_RC1= !SV3.FB & SV2.FB & SV1.FB & !SV0.FB ;
CLR_RC2= SV3.FB & !SV2.FB & !SV1.FB & SV0.FB ;
CLR_RC3= !SV3.FB & !SV2.FB & SV1.FB & SV0.FB ;
DONE= SV3.FB & !SV2.FB & SV1.FB & SV0.FB ;
E_FIFO_OE= SV3.FB & !SV2.FB & !SV1.FB & !SV0.FB # !SV3.FB & !SV2.FB &
SV1.FB & !SV0.FB # SV3.FB & SV2.FB & SV1.FB & !SV0.FB # !SV3.FB & SV2.FB
& !SV1.FB & SV0.FB ;
RD_E_FIFO= SV3.FB & !SV2.FB & !SV1.FB & !SV0.FB # !SV3.FB & !SV2.FB &
SV1.FB & !SV0.FB # SV3.FB & SV2.FB & SV1.FB & !SV0.FB # !SV3.FB & SV2.FB
& !SV1.FB & SV0.FB ;
SRC_ERR0= SV3.FB & SV2.FB & SV1.FB & SV0.FB ;
WRT_P_FIFO= SV3.FB & SV2.FB & !SV1.FB & !SV0.FB # !SV3.FB & SV2.FB & SV1.FB
& !SV0.FB # SV3.FB & !SV2.FB & !SV1.FB & SV0.FB # !SV3.FB & !SV2.FB &
SV1.FB & SV0.FB ;
WRT_P_REG= SV3.FB & !SV2.FB & !SV1.FB & !SV0.FB # !SV3.FB & !SV2.FB &
SV1.FB & !SV0.FB # SV3.FB & SV2.FB & SV1.FB & !SV0.FB # !SV3.FB & SV2.FB
& !SV1.FB & SV0.FB ;
"Logic Equations
EQUATIONS
TSI_WAIT = E_FIFO_EMPTY ;

END L2A_SM

```

#### 4.6.3 RDQ\_SM

```
" D:\CDF\SRC\RSM\RDQ_SM.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.2
" Thu Nov 13 15:26:09 1997

" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are area optimized.

MODULE RDQ_SM

DECLARATIONS
"clock name
    CLK PIN;

"Input variables
    DUMP_Q PIN;
    INC_DIG PIN;
    RD_BUSY PIN;

"Output variables
    RDQ0 PIN ISTYPE 'com';
    RDQ1 PIN ISTYPE 'com';

"State variables
    SV0 PIN ISTYPE 'reg';
    SV1 PIN ISTYPE 'reg';

"Vectors
DECLARATIONS
    RDQ=[  
        RDQ1,  
        RDQ0  
    ];

"State Register assignment
DECLARATIONS
    sreg=[ SV0,SV1];

EQUATIONS
    sreg.clk=CLK;

DECLARATIONS
    STATE0=[0, 0];
    STATE1=[0, 1];
    STATE2=[1, 0];

EQUATIONS
```

```
SV0 := !DUMP_Q & !INC_DIG & !SV0.FB & SV1.FB # !DUMP_Q & RD_BUSY & SV0.FB &
!SV1.FB ;
SV1 := !DUMP_Q & INC_DIG & !SV0.FB ;
RDQ=( !SV0.FB & !SV1.FB & ([0,0] ) )#( !SV0.FB & SV1.FB & ([0,1] ) )#
( SV0.FB & !SV1.FB & ([1,0] ) );
END RDQ_SM
```

#### 4.6.4 READ\_Q

```
" D:\CDF\SRC\RSM\READ_Q.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.2
" Mon Dec 29 15:33:20 1997

" This Abel code was generated using:
" binary encoded state assignment with structured code format.
" Minimization is enabled, implied else is enabled,
" and outputs are area optimized.
```

```
MODULE READ_Q
```

```
DECLARATIONS
```

```
"clock name
    CLK PIN;
```

```
"Input variables
```

```
    DN PIN;
    DUMP_Q PIN;
    UP PIN;
```

```
"Output variables
```

```
    L2Q0 PIN ISTYPE 'com';
    L2Q1 PIN ISTYPE 'com';
    L2Q2 PIN ISTYPE 'com';
    L2Q3 PIN ISTYPE 'com';
```

```
"State variables
```

```
    SV0 PIN ISTYPE 'reg';
    SV1 PIN ISTYPE 'reg';
    SV2 PIN ISTYPE 'reg';
    SV3 PIN ISTYPE 'reg';
```

```
"Vectors
```

```
DECLARATIONS
```

```
    L2Q=[  
        L2Q3,  
        L2Q2,  
        L2Q1,  
        L2Q0  
    ];
```

```
"State Register assignment
```

```
DECLARATIONS
```

```
    sreg=[ SV0,SV1,SV2,SV3];
```

```
EQUATIONS
```

```
    sreg.clk=CLK;
```

```
DECLARATIONS
```

```
    STATE0=[0, 0, 0, 0];
```

```

STATE1=[0, 0, 0, 1];
STATE2=[0, 0, 1, 0];
STATE3=[0, 0, 1, 1];
STATE4=[0, 1, 0, 0];
STATE5=[0, 1, 0, 1];
STATE6=[0, 1, 1, 0];
STATE7=[0, 1, 1, 1];
STATE8=[1, 0, 0, 0];

"Asynchronous Reset
EQUATIONS
[SV0, SV1, SV2, SV3].ar = DUMP_Q ;

state_diagram sreg;

state STATE0:
    L2Q=[1,1,1,1];
    IF ( !DN & !UP # UP & DN ) THEN STATE0;
    IF ( DN & !UP ) THEN STATE5;
    IF ( UP & !DN ) THEN STATE1;

state STATE1:
    L2Q=[0,1,1,1];
    IF ( DN & !UP ) THEN STATE0;
    IF ( !DN & !UP # UP & DN ) THEN STATE1;
    IF ( UP & !DN ) THEN STATE2;

state STATE2:
    L2Q=[0,0,1,1];
    IF ( DN & !UP ) THEN STATE1;
    IF ( !DN & !UP # UP & DN ) THEN STATE2;
    IF ( UP & !DN ) THEN STATE3;

state STATE3:
    L2Q=[0,0,0,1];
    IF ( DN & !UP ) THEN STATE2;
    IF ( !DN & !UP # UP & DN ) THEN STATE3;
    IF ( UP & !DN ) THEN STATE4;

state STATE4:
    L2Q=[0,0,0,0];
    IF ( DN & !UP ) THEN STATE3;
    IF ( UP # !DN ) THEN STATE4;

state STATE5:
    L2Q=[1,1,1,1];
    IF ( !UP & !DN # DN & UP ) THEN STATE5;
    IF ( UP & !DN ) THEN STATE0;
    IF ( DN & !UP ) THEN STATE6;

state STATE6:
    L2Q=[1,1,1,1];
    IF ( !UP & !DN # DN & UP ) THEN STATE6;
    IF ( UP & !DN ) THEN STATE5;
    IF ( DN & !UP ) THEN STATE7;

state STATE7:
    L2Q=[1,1,1,1];
    IF ( !UP & !DN # DN & UP ) THEN STATE7;
    IF ( UP & !DN ) THEN STATE6;
    IF ( DN & !UP ) THEN STATE8;

```

```
state STATE8:  
    L2Q=[1,1,1,1];  
    IF ( UP & !DN ) THEN STATE7;  
    IF ( DN # !UP ) THEN STATE8;  
  
END READ_Q
```

#### 4.6.5 READ\_SM6

```
" D:\CDF\SRC\RSM\READ_SM6.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.2
" Thu Nov 13 15:28:22 1997

" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are area optimized.
```

```
MODULE READ_SM6
```

##### DECLARATIONS

```
"clock name
    CLK PIN;
```

##### "Input variables

```
BX PIN;
CALI PIN;
DIGEQ PIN;
DO_IT PIN;
DUMP_Q PIN;
GAP PIN;
L1_PFE PIN;
L1_PTR0 PIN;
L1_PTR1 PIN;
L1_PTR2 PIN;
L1_PTR3 PIN;
L2R0 PIN;
L2R1 PIN;
L2R2 PIN;
L2R3 PIN;
MASK0 PIN;
MASK1 PIN;
MASK2 PIN;
MASK3 PIN;
P_FE PIN;
RC0 PIN;
RC1 PIN;
RC2 PIN;
RC3 PIN;
RD_BUSY PIN;
SCAN PIN;
SCAN_BUSY PIN;
```

##### "Output variables

```
CLR_DIG PIN ISTYPE 'com';
CLR_L2R0 PIN ISTYPE 'com';
CLR_L2R1 PIN ISTYPE 'com';
CLR_L2R2 PIN ISTYPE 'com';
CLR_L2R3 PIN ISTYPE 'com';
```

```

CLR_RC0 PIN ISTYPE 'com';
CLR_RC1 PIN ISTYPE 'com';
CLR_RC2 PIN ISTYPE 'com';
CLR_RC3 PIN ISTYPE 'com';
CLR_SCAN PIN ISTYPE 'com';
CMD0 PIN ISTYPE 'com';
CMD1 PIN ISTYPE 'com';
CMD2 PIN ISTYPE 'com';
CMD3 PIN ISTYPE 'com';
EN_STRB PIN ISTYPE 'com';
FCMD0 PIN ISTYPE 'com';
FCMD1 PIN ISTYPE 'com';
FCMD2 PIN ISTYPE 'com';
FCMD3 PIN ISTYPE 'com';
FCMD4 PIN ISTYPE 'com';
INC_DIG PIN ISTYPE 'com';
INC_L2Q PIN ISTYPE 'com';
JOINED PIN ISTYPE 'com';
L1_DONE PIN ISTYPE 'com';
LATCH_VRB PIN ISTYPE 'com';
P_FIFO_OE PIN ISTYPE 'com';
PIPE_RD2 PIN ISTYPE 'com';
RD_BUNCH PIN ISTYPE 'com';
RD_L1PF PIN ISTYPE 'com';
RD_PF PIN ISTYPE 'com';
RD_PIPE PIN ISTYPE 'com';
RD_R_A PIN ISTYPE 'com';
SCAN_EID_OE PIN ISTYPE 'com';
SEL_BUF PIN ISTYPE 'com';
SEPP PIN ISTYPE 'com';
SET_RC0 PIN ISTYPE 'com';
SET_RC1 PIN ISTYPE 'com';
SET_RC2 PIN ISTYPE 'com';
SET_RC3 PIN ISTYPE 'com';
SET_SCAN PIN ISTYPE 'com';
SUB0 PIN ISTYPE 'com';
SUB1 PIN ISTYPE 'com';
SUB2 PIN ISTYPE 'com';
SUB3 PIN ISTYPE 'com';
WRITE_EF PIN ISTYPE 'com';

```

"Logic variables

```

CRT_L2R PIN ISTYPE 'com';
DO_L2R PIN ISTYPE 'com';
ERA PIN ISTYPE 'com';
L1_MSK0 PIN ISTYPE 'com';
L1_MSK1 PIN ISTYPE 'com';
L1_MSK2 PIN ISTYPE 'com';
L1_MSK3 PIN ISTYPE 'com';
P_L2R0 PIN ISTYPE 'com';
P_L2R1 PIN ISTYPE 'com';
P_L2R2 PIN ISTYPE 'com';
P_L2R3 PIN ISTYPE 'com';

```

"State variables

```
SV0 PIN ISTYPE 'reg';
SV1 PIN ISTYPE 'reg';
SV2 PIN ISTYPE 'reg';
SV3 PIN ISTYPE 'reg';
SV4 PIN ISTYPE 'reg';
SV5 PIN ISTYPE 'reg';
SV6 PIN ISTYPE 'reg';
SV7 PIN ISTYPE 'reg';
SV8 PIN ISTYPE 'reg';
```

"Vectors

DECLARATIONS

```
CLR_L2R=[  
    CLR_L2R3,  
    CLR_L2R2,  
    CLR_L2R1,  
    CLR_L2R0  
];
```

```
CMD=[  
    CMD3,  
    CMD2,  
    CMD1,  
    CMD0  
];
```

```
FCMD=[  
    FCMD4,  
    FCMD3,  
    FCMD2,  
    FCMD1,  
    FCMD0  
];
```

```
L1_MSK=[  
    L1_MSK3,  
    L1_MSK2,  
    L1_MSK1,  
    L1_MSK0  
];
```

```
L1_PTR=[  
    L1_PTR3,  
    L1_PTR2,  
    L1_PTR1,  
    L1_PTR0  
];
```

```
L2R=[  
    L2R3,  
    L2R2,  
    L2R1,  
    L2R0  
];
```

```

MASK=[  

    MASK3,  

    MASK2,  

    MASK1,  

    MASK0  

];  
  

P_L2R=[  

    P_L2R3,  

    P_L2R2,  

    P_L2R1,  

    P_L2R0  

];  
  

RC=[  

    RC3,  

    RC2,  

    RC1,  

    RC0  

];  
  

SET_RC=[  

    SET_RC3,  

    SET_RC2,  

    SET_RC1,  

    SET_RC0  

];  
  

SUB=[  

    SUB3,  

    SUB2,  

    SUB1,  

    SUB0  

];
";  
  

"State Register assignment  

DECLARATIONS  

sreg=[ SV0,SV1,SV2,SV3,SV4,SV5];  
  

EQUATIONS  

sreg.clk=CLK;  
  

DECLARATIONS  

STATE0=[0, 0, 0, 0, 0, 0];  

STATE1=[0, 0, 0, 0, 0, 1];  

STATE2=[0, 0, 0, 0, 1, 0];  

STATE3=[0, 0, 0, 0, 1, 1];  

STATE4=[0, 0, 0, 1, 0, 0];  

STATE5=[0, 0, 0, 1, 0, 1];  

STATE6=[0, 0, 0, 1, 1, 0];  

STATE7=[0, 0, 0, 1, 1, 1];  

STATE8=[0, 0, 1, 0, 0, 0];  

STATE9=[0, 0, 1, 0, 0, 1];  

STATE10=[0, 0, 1, 0, 1, 0];  

STATE11=[0, 0, 1, 0, 1, 1];

```

```

STATE12=[0, 0, 1, 1, 0, 0];
STATE13=[0, 0, 1, 1, 0, 1];
STATE14=[0, 0, 1, 1, 1, 0];
STATE15=[0, 0, 1, 1, 1, 1];
STATE16=[0, 1, 0, 0, 0, 0];
STATE17=[0, 1, 0, 0, 0, 1];
STATE18=[0, 1, 0, 0, 1, 0];
STATE19=[0, 1, 0, 0, 1, 1];
STATE25=[0, 1, 0, 1, 0, 0];
STATE26=[0, 1, 0, 1, 0, 1];
STATE27=[0, 1, 0, 1, 1, 0];
STATE28=[0, 1, 0, 1, 1, 1];
STATE29=[0, 1, 1, 0, 0, 0];
STATE30=[0, 1, 1, 0, 0, 1];
STATE31=[0, 1, 1, 0, 1, 0];
STATE32=[0, 1, 1, 0, 1, 1];
STATE33=[0, 1, 1, 1, 0, 0];
STATE34=[0, 1, 1, 1, 0, 1];
STATE81=[0, 1, 1, 1, 1, 0];
STATE82=[0, 1, 1, 1, 1, 1];
STATE83=[1, 0, 0, 0, 0, 0];
STATE84=[1, 0, 0, 0, 0, 1];
STATE85=[1, 0, 0, 0, 1, 0];
STATE86=[1, 0, 0, 0, 1, 1];
STATE91=[1, 0, 0, 1, 0, 0];
STATE92=[1, 0, 0, 1, 0, 1];
STATE93=[1, 0, 0, 1, 1, 0];
STATE94=[1, 0, 0, 1, 1, 1];
STATE119=[1, 0, 1, 0, 0, 0];
STATE120=[1, 0, 1, 0, 0, 1];
STATE131=[1, 0, 1, 0, 1, 0];
STATE132=[1, 0, 1, 0, 1, 1];
STATE133=[1, 0, 1, 1, 0, 0];

```

## EQUATIONS

```

SV0 := !CALI & SCAN & !SCAN_BUSY & SV0.FB & !SV1.FB & !SV3.FB & SV4.FB &
SV5.FB # GAP & SCAN & !SCAN_BUSY & SV0.FB & !SV1.FB & !SV3.FB & SV4.FB &
SV5.FB # !SV0.FB & SV1.FB & !SV2.FB & SV4.FB & SV5.FB # !RC0 & !RC1 & !
RC2 & !RC3 & !SV0.FB & SV1.FB & SV2.FB & SV3.FB & !SV4.FB # !L2R0 & !RC1 &
!RC2 & !RC3 & !SV0.FB & SV1.FB & SV2.FB & SV3.FB & !SV4.FB # !RC0 & !L2R1 &
!RC2 & !RC3 & !SV0.FB & SV1.FB & SV2.FB & SV3.FB & !SV4.FB # !L2R0 & !L2R1
& !RC2 & !RC3 & !SV0.FB & SV1.FB & SV2.FB & SV3.FB & !SV4.FB # !RC0 & !RC1
& !L2R2 & !RC3 & !SV0.FB & SV1.FB & SV2.FB & SV3.FB & !SV4.FB # !L2R0 & !
RC1 & !L2R2 & !RC3 & !SV0.FB & SV1.FB & SV2.FB & SV3.FB & !SV4.FB # !RC0 &
!L2R1 & !L2R2 & !RC3 & !SV0.FB & SV1.FB & SV2.FB & SV3.FB & !SV4.FB # !L2R0
& !L2R1 & !L2R2 & !RC3 & !SV0.FB & SV1.FB & SV2.FB & SV3.FB & !SV4.FB # !
RC0 & !RC1 & !RC2 & !L2R3 & !SV0.FB & SV1.FB & SV2.FB & SV3.FB & !SV4.FB #
!L2R0 & !RC1 & !RC2 & !L2R3 & !SV0.FB & SV1.FB & SV2.FB & SV3.FB & !SV4.FB #
!RC0 & !L2R1 & !RC2 & !L2R3 & !SV0.FB & SV1.FB & SV2.FB & SV3.FB & !SV4.FB
# !L2R0 & !L2R1 & !RC2 & !L2R3 & !SV0.FB & SV1.FB & SV2.FB & SV3.FB & !
SV4.FB # !RC0 & !RC1 & !L2R2 & !L2R3 & !SV0.FB & SV1.FB & SV2.FB & SV3.FB &
!SV4.FB # !L2R0 & !RC1 & !L2R2 & !L2R3 & !SV0.FB & SV1.FB & SV2.FB &
SV3.FB & !SV4.FB # !RC0 & !L2R1 & !L2R2 & !L2R3 & !SV0.FB & SV1.FB & SV2.FB
& SV3.FB & !SV4.FB # !L2R0 & !L2R1 & !L2R2 & !L2R3 & !SV0.FB & SV1.FB &

```

```

SV2.FB & SV3.FB & !SV4.FB # SV0.FB & !SV1.FB & SV2.FB & !SV3.FB & SV4.FB #
!SV0.FB & SV1.FB & SV3.FB & SV5.FB # SV0.FB & !SV1.FB & !SV2.FB & SV3.FB
& !SV4.FB # SV0.FB & !SV1.FB & !SV2.FB & SV3.FB & !SV5.FB # !SV0.FB &
SV2.FB & SV3.FB & !SV4.FB & SV5.FB # SV0.FB & !SV1.FB & SV2.FB & !SV3.FB &
!SV5.FB # SV0.FB & !SV1.FB & !SV2.FB & !SV4.FB & !SV5.FB # !CALI &
SCAN_BUSY & !DO_L2R & SV0.FB & !SV1.FB & !SV3.FB & SV4.FB & SV5.FB # GAP &
SCAN_BUSY & !DO_L2R & SV0.FB & !SV1.FB & !SV3.FB & SV4.FB & SV5.FB # !CALI
& !SCAN & P_FE & !DO_L2R & SV0.FB & !SV1.FB & !SV3.FB & SV4.FB & SV5.FB #
GAP & !SCAN & P_FE & !DO_L2R & SV0.FB & !SV1.FB & !SV3.FB & SV4.FB & SV5.FB
# !CALI & !DO_IT & SV0.FB & !SV1.FB & !SV3.FB & SV4.FB & SV5.FB # GAP & !
DO_IT & SV0.FB & !SV1.FB & !SV3.FB & SV4.FB & SV5.FB # SV0.FB & !SV1.FB &
SV3.FB & !SV4.FB & !SV5.FB # DUMP_Q;

SV1 := !DUMP_Q & !RD_BUSY & !SV0.FB & !SV1.FB & SV2.FB & SV3.FB & SV4.FB &
SV5.FB # !DUMP_Q & CRT_L2R & !SV0.FB & !SV1.FB & SV2.FB & SV3.FB & SV4.FB &
SV5.FB # !DUMP_Q & SV0.FB & !SV1.FB & SV2.FB & !SV3.FB & !SV4.FB & SV5.FB
# !DUMP_Q & !SV0.FB & !SV1.FB & !SV2.FB & SV3.FB & !SV4.FB & SV5.FB # !
DUMP_Q & ERA & !SV0.FB & !SV1.FB & SV2.FB & SV3.FB & SV4.FB & SV5.FB # !
DUMP_Q & SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB & SV4.FB & !SV5.FB # !DUMP_Q
& !SV0.FB & SV1.FB & !SV2.FB & !SV5.FB # !DUMP_Q & !RC3 & !RC2 & !RC1 &
L2R0 & RC0 & !SV0.FB & SV1.FB & !SV5.FB # !DUMP_Q & !L2R3 & !RC2 & !RC1 &
L2R0 & RC0 & !SV0.FB & SV1.FB & !SV5.FB # !DUMP_Q & !RC3 & !L2R2 & !RC1 &
L2R0 & RC0 & !SV0.FB & SV1.FB & !SV5.FB # !DUMP_Q & !L2R3 & !L2R2 & !RC1 &
L2R0 & RC0 & !SV0.FB & SV1.FB & !SV5.FB # !DUMP_Q & !RC3 & !L2R1 &
L2R0 & RC0 & !SV0.FB & SV1.FB & !SV5.FB # !DUMP_Q & !L2R3 & !RC2 & !L2R1 &
L2R0 & RC0 & !SV0.FB & SV1.FB & !SV5.FB # !DUMP_Q & !RC3 & !L2R2 & !L2R1 &
L2R0 & RC0 & !SV0.FB & SV1.FB & !SV5.FB # !DUMP_Q & !L2R3 & !L2R2 & !L2R1 &
L2R0 & RC0 & !SV0.FB & SV1.FB & !SV5.FB # !DUMP_Q & L2R3 & RC3 & !SV0.FB &
SV1.FB & !SV5.FB # !DUMP_Q & !RC3 & L2R2 & RC2 & !SV0.FB & SV1.FB & !
SV5.FB # !DUMP_Q & !L2R3 & L2R2 & RC2 & !SV0.FB & SV1.FB & !SV5.FB # !
DUMP_Q & !RC3 & !RC2 & L2R1 & RC1 & !SV0.FB & SV1.FB & !SV5.FB # !DUMP_Q &
!L2R3 & !RC2 & L2R1 & RC1 & !SV0.FB & SV1.FB & !SV5.FB # !DUMP_Q & !RC3 & !
L2R2 & L2R1 & RC1 & !SV0.FB & SV1.FB & !SV5.FB # !DUMP_Q & !L2R3 & !L2R2 &
L2R1 & RC1 & !SV0.FB & SV1.FB & !SV5.FB # !DUMP_Q & !SV0.FB & SV1.FB & !
SV3.FB & !SV4.FB # !DUMP_Q & !SV0.FB & SV1.FB & !SV3.FB & !SV5.FB # !
DUMP_Q & !SV0.FB & SV1.FB & SV2.FB & !SV3.FB # !DUMP_Q & !CALI & DO_IT &
SCAN_BUSY & DO_L2R & SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB & SV4.FB # !
DUMP_Q & GAP & DO_IT & SCAN_BUSY & DO_L2R & SV0.FB & !SV1.FB & !SV2.FB & !
SV3.FB & SV4.FB # !DUMP_Q & !CALI & DO_IT & !SCAN & P_FE & DO_L2R & SV0.FB
& !SV1.FB & !SV2.FB & !SV3.FB & !SV4.FB # !DUMP_Q & GAP & DO_IT & !SCAN &
P_FE & DO_L2R & SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB & SV4.FB # !DUMP_Q &
CALI & !GAP & SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB & SV4.FB # !DUMP_Q & !
CALI & DO_IT & !SCAN & !P_FE & !SCAN_BUSY & SV0.FB & !SV1.FB & !SV2.FB &
SV3.FB & SV4.FB # !DUMP_Q & GAP & DO_IT & !SCAN & !P_FE & !SCAN_BUSY &
SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB & SV4.FB # !DUMP_Q & !SV0.FB & SV1.FB
& SV4.FB & !SV5.FB ;

SV2 := !SV0.FB & !SV1.FB & SV2.FB & !SV5.FB # RD_BUSY & !ERA & !CRT_L2R & !
SV0.FB & !SV1.FB & SV3.FB & SV4.FB & SV5.FB # !SV1.FB & !SV2.FB & SV3.FB &
SV4.FB & SV5.FB # !RC3 & !RC2 & !RC1 & L2R0 & RC0 & !SV0.FB & SV2.FB & !
SV5.FB # !L2R3 & !RC2 & !RC1 & L2R0 & RC0 & !SV0.FB & SV2.FB & !SV5.FB #
!RC3 & !L2R2 & !RC1 & L2R0 & RC0 & !SV0.FB & SV2.FB & !SV5.FB # !L2R3 & !
L2R2 & !RC1 & L2R0 & RC0 & !SV0.FB & SV2.FB & !SV5.FB # !RC3 & !RC2 & !L2R1 &
L2R0 & RC0 & !SV0.FB & SV2.FB & !SV5.FB # !L2R3 & !RC2 & !L2R1 & L2R0 &
RC0 & !SV0.FB & SV2.FB & !SV5.FB # !RC3 & !L2R2 & !L2R1 & L2R0 & RC0 & !

```

```

SV0.FB & SV2.FB & !SV5.FB # !L2R3 & !L2R2 & !L2R1 & L2R0 & RC0 & !SV0.FB &
SV2.FB & !SV5.FB # L2R3 & RC3 & !SV0.FB & SV2.FB & !SV5.FB # !RC3 & L2R2
& RC2 & !SV0.FB & SV2.FB & !SV5.FB # !L2R3 & L2R2 & RC2 & !SV0.FB & SV2.FB
& !SV5.FB # !RC3 & !RC2 & L2R1 & RC1 & !SV0.FB & SV2.FB & !SV5.FB # !L2R3
& !RC2 & L2R1 & RC1 & !SV0.FB & SV2.FB & !SV5.FB # !RC3 & !L2R2 & L2R1 &
RC1 & !SV0.FB & SV2.FB & !SV5.FB # !L2R3 & !L2R2 & L2R1 & RC1 & !SV0.FB &
SV2.FB & !SV5.FB # !SV0.FB & SV2.FB & !SV3.FB # !CALI & SCAN_BUSY &
DO_L2R & SV0.FB & !SV1.FB & !SV2.FB & SV4.FB & SV5.FB # GAP & SCAN_BUSY &
DO_L2R & SV0.FB & !SV1.FB & !SV2.FB & SV4.FB & SV5.FB # !CALI & !SCAN &
P_FE & DO_L2R & SV0.FB & !SV1.FB & !SV2.FB & SV4.FB & SV5.FB # GAP & !SCAN
& P_FE & DO_L2R & SV0.FB & !SV1.FB & !SV2.FB & SV4.FB & SV5.FB # CALI & !
GAP & SV0.FB & !SV1.FB & !SV2.FB & SV4.FB & SV5.FB # !CALI & !SCAN & !P_FE
& !SCAN_BUSY & SV0.FB & !SV1.FB & !SV2.FB & SV4.FB & SV5.FB # GAP & !SCAN &
!P_FE & !SCAN_BUSY & SV0.FB & !SV1.FB & !SV2.FB & SV4.FB & SV5.FB # !
SV0.FB & SV2.FB & SV4.FB & !SV5.FB # !SV0.FB & !SV1.FB & SV2.FB & !SV4.FB
# !SV1.FB & SV2.FB & !SV3.FB & !SV4.FB & !SV5.FB # SV0.FB & !SV1.FB & !
SV3.FB & !SV4.FB & !SV5.FB # !CALI & SCAN_BUSY & !L1_PFE & SV0.FB & !SV1.FB
& !SV2.FB & SV4.FB & SV5.FB # GAP & SCAN_BUSY & !L1_PFE & SV0.FB & !SV1.FB
& !SV2.FB & SV4.FB & SV5.FB # !CALI & !SCAN & P_FE & !L1_PFE & SV0.FB & !
SV1.FB & !SV2.FB & SV4.FB & SV5.FB # GAP & !SCAN & P_FE & !L1_PFE & SV0.FB
& !SV1.FB & !SV2.FB & SV4.FB & SV5.FB # !CALI & !DO_IT & SV0.FB & !SV1.FB &
!SV2.FB & SV4.FB & SV5.FB # GAP & !DO_IT & SV0.FB & !SV1.FB & !SV2.FB &
SV4.FB & SV5.FB # !DO_IT & !SV1.FB & SV2.FB & !SV4.FB & !SV5.FB # DUMP_Q
;

```

```

SV3 := !CRT_L2R & !SV0.FB & !SV1.FB & !SV2.FB & !SV4.FB & !SV5.FB # !SV0.FB
& !SV1.FB & !SV3.FB & SV4.FB & SV5.FB # !BX & !SV0.FB & !SV1.FB & SV2.FB &
!SV3.FB & SV4.FB # !SV0.FB & !SV1.FB & SV3.FB & !SV5.FB # !CRT_L2R &
RD_BUSY & !SV0.FB & !SV1.FB & SV2.FB & SV4.FB & SV5.FB # SV0.FB & !SV1.FB &
!SV2.FB & SV4.FB & !SV5.FB # !SV0.FB & !SV2.FB & SV3.FB & !SV5.FB # !
SV0.FB & SV1.FB & SV2.FB & !SV3.FB & !SV4.FB # !SV0.FB & SV1.FB & SV2.FB &
!SV3.FB & !SV5.FB # !SV0.FB & SV2.FB & !SV3.FB & SV4.FB & SV5.FB # !CALI
& SCAN_BUSY & DO_L2R & !SV1.FB & !SV2.FB & !SV3.FB & SV4.FB & SV5.FB # GAP
& SCAN_BUSY & DO_L2R & !SV1.FB & !SV2.FB & !SV3.FB & SV4.FB & SV5.FB # !
CALI & !SCAN & P_FE & DO_L2R & !SV1.FB & !SV2.FB & !SV3.FB & SV4.FB & SV5.FB
# GAP & !SCAN & P_FE & DO_L2R & !SV1.FB & !SV2.FB & !SV3.FB & SV4.FB &
SV5.FB # CALI & !GAP & !SV1.FB & !SV2.FB & !SV3.FB & SV4.FB & SV5.FB # !
CALI & !SCAN & !P_FE & !SCAN_BUSY & !SV1.FB & !SV2.FB & !SV3.FB & SV4.FB &
SV5.FB # GAP & !SCAN & !P_FE & !SCAN_BUSY & !SV1.FB & !SV2.FB & !SV3.FB &
SV4.FB & SV5.FB # !SV0.FB & SV3.FB & SV4.FB & !SV5.FB # !SV0.FB & SV1.FB
& !SV2.FB & SV3.FB & !SV4.FB # SV0.FB & !SV1.FB & !SV2.FB & SV3.FB # !
SV1.FB & !SV2.FB & SV3.FB & !SV5.FB # !CALI & !DO_IT & !SV1.FB & !SV2.FB &
!SV3.FB & SV4.FB & SV5.FB # GAP & !DO_IT & !SV1.FB & !SV2.FB & !SV3.FB &
SV4.FB & SV5.FB # !DO_IT & !SV1.FB & SV3.FB & !SV4.FB & !SV5.FB # DUMP_Q
;

```

```

SV4 := !DUMP_Q & !CRT_L2R & !SV0.FB & !SV2.FB & !SV3.FB & !SV4.FB # !DUMP_Q
& DIGEQ & !SV0.FB & !SV1.FB & !SV3.FB & !SV4.FB & SV5.FB # !DUMP_Q &
CRT_L2R & !SV0.FB & !SV1.FB & !SV3.FB & !SV4.FB & SV5.FB # !DUMP_Q & !
SV0.FB & !SV1.FB & SV4.FB & !SV5.FB # !DUMP_Q & RD_BUSY & !ERA & !CRT_L2R
&
!SV0.FB & !SV1.FB & SV2.FB & SV3.FB & SV4.FB # !DUMP_Q & !SV0.FB & !SV2.FB
& !SV3.FB & !SV4.FB & SV5.FB # !DUMP_Q & SV0.FB & !SV1.FB & SV2.FB & !
SV3.FB & !SV4.FB & SV5.FB # !DUMP_Q & !SV0.FB & !SV1.FB & !SV2.FB & !SV4.FB
& SV5.FB # !DUMP_Q & !SV0.FB & SV1.FB & !SV2.FB & !SV3.FB & !SV4.FB # !

```



SCAN & P\_FE & !DO\_L2R & SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB & SV4.FB ;  
  
 SV5 := !DUMP\_Q & CRT\_L2R & !SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB & !SV5.FB #  
     !DUMP\_Q & !BX & !SV0.FB & !SV1.FB & !SV2.FB & SV3.FB & !SV5.FB # !DUMP\_Q  
     & !SV0.FB & !SV1.FB & !SV2.FB & SV4.FB & !SV5.FB # !DUMP\_Q & !DIGEQ &  
     !SV0.FB & !SV1.FB & SV2.FB & !SV3.FB & !SV4.FB # !DUMP\_Q & CRT\_L2R & !SV0.FB  
     & !SV1.FB & SV2.FB & !SV3.FB & !SV4.FB # !DUMP\_Q & !SV0.FB & !SV1.FB &  
     SV2.FB & !SV4.FB & !SV5.FB # !DUMP\_Q & RD\_BUSY & !ERA & !SV0.FB & !SV1.FB &  
     SV2.FB & SV3.FB & SV4.FB # !DUMP\_Q & !SV0.FB & !SV1.FB & !SV2.FB & SV3.FB &  
     SV4.FB & SV5.FB # !DUMP\_Q & CRT\_L2R & !SV0.FB & !SV1.FB & SV2.FB & SV3.FB  
     & !SV4.FB # !DUMP\_Q & !SV0.FB & !SV1.FB & !SV2.FB & SV3.FB & !SV4.FB &  
     SV5.FB # !DUMP\_Q & !RD\_BUSY & !SV0.FB & !SV2.FB & SV4.FB & !SV5.FB # !  
     DUMP\_Q & !SV0.FB & SV1.FB & !SV2.FB & !SV4.FB & !SV5.FB # !DUMP\_Q & L2R3 &  
     RC3 & !SV0.FB & SV2.FB & SV3.FB & !SV5.FB # !DUMP\_Q & !RC3 & !RC2 & L2R1 &  
     RC1 & !SV0.FB & SV2.FB & SV3.FB & !SV5.FB # !DUMP\_Q & !L2R3 & !RC2 & L2R1 &  
     RC1 & !SV0.FB & SV2.FB & SV3.FB & !SV5.FB # !DUMP\_Q & !RC3 & !L2R2 & L2R1  
     & RC1 & !SV0.FB & SV2.FB & SV3.FB & !SV5.FB # !DUMP\_Q & !L2R3 & !L2R2 &  
     L2R1 & RC1 & !SV0.FB & SV2.FB & SV3.FB & !SV5.FB # !DUMP\_Q & CALI & !GAP &  
     SV0.FB & !SV1.FB & !SV2.FB & SV4.FB & SV5.FB # !DUMP\_Q & !SV0.FB & SV3.FB &  
     SV4.FB & !SV5.FB # !DUMP\_Q & !SV0.FB & SV1.FB & !SV2.FB & SV4.FB & SV5.FB  
     # !DUMP\_Q & !SV0.FB & SV1.FB & !SV2.FB & SV3.FB & SV4.FB # !RC0 & !RC1 &  
     !RC2 & !RC3 & !DUMP\_Q & !SV0.FB & SV2.FB & SV3.FB & !SV5.FB # !L2R0 & !RC1  
     & !RC2 & !RC3 & !DUMP\_Q & !SV0.FB & SV2.FB & SV3.FB & !SV5.FB # !RC0 &  
     L2R1 & !RC2 & !RC3 & !DUMP\_Q & !SV0.FB & SV2.FB & SV3.FB & !SV5.FB # !  
     RC0 & !RC1 & !L2R2 & !RC3 & !DUMP\_Q & !SV0.FB & SV2.FB & SV3.FB & !SV5.FB  
     # !L2R0 & !RC1 & !L2R2 & !RC3 & !DUMP\_Q & !SV0.FB & SV2.FB & SV3.FB &  
     !SV5.FB # !L2R0 & !L2R1 & !L2R2 & !RC3 & !DUMP\_Q & !SV0.FB & SV2.FB &  
     & SV3.FB & !SV5.FB # !RC0 & !RC1 & !RC2 & !L2R3 & !DUMP\_Q & !SV0.FB &  
     SV2.FB & SV3.FB & !SV5.FB # !L2R0 & !L2R1 & !RC2 & !L2R3 & !DUMP\_Q &  
     SV0.FB & SV2.FB & SV3.FB & !SV5.FB # !RC0 & !RC1 & !L2R2 & !L2R3 &  
     !DUMP\_Q & !SV0.FB & SV2.FB & SV3.FB & !SV5.FB # !RC0 & !L2R1 & !L2R2 &  
     !L2R3 & !DUMP\_Q & !SV0.FB & SV2.FB & SV3.FB & !SV5.FB # !DUMP\_Q & !SV0.FB &  
     SV1.FB & SV2.FB & SV3.FB & !SV4.FB & SV5.FB # !DUMP\_Q & SV0.FB & !SV1.FB &  
     SV2.FB & !SV3.FB & SV4.FB # !DUMP\_Q & DO\_IT & !SV1.FB & SV2.FB & SV3.FB &  
     !SV4.FB & !SV5.FB # !DUMP\_Q & SV0.FB & !SV1.FB & !SV2.FB & SV3.FB & !SV4.FB  
     & !SV5.FB # !DUMP\_Q & !BX & !SV1.FB & !SV2.FB & SV3.FB & !SV4.FB & !SV5.FB # !  
     DUMP\_Q & !CALI & DO\_IT & SCAN\_BUSY & !DO\_L2R & SV0.FB & !SV1.FB & !SV2.FB  
     &  
     !  
     SV4.FB & SV5.FB # !DUMP\_Q & GAP & DO\_IT & SCAN\_BUSY & !DO\_L2R & SV0.FB &  
     !  
     SV1.FB & !SV2.FB & SV4.FB & SV5.FB # !DUMP\_Q & !CALI & DO\_IT & !SCAN & P\_FE  
     & !DO\_L2R & SV0.FB & !SV1.FB & !SV2.FB & SV4.FB & SV5.FB # !DUMP\_Q & GAP &  
     DO\_IT & !SCAN & P\_FE & !DO\_L2R & SV0.FB & !SV1.FB & !SV2.FB & SV4.FB &  
     SV5.FB ;

"State Register assignment

DECLARATIONS

sreg1=[ SV6,SV7,SV8];



CLR\_RC0= !SV5.FB & !SV4.FB & !SV3.FB & SV2.FB & SV1.FB & !SV0.FB ;

CLR\_RC1= SV5.FB & SV4.FB & !SV3.FB & SV2.FB & SV1.FB & !SV0.FB ;

CLR\_RC2= !SV5.FB & SV4.FB & !SV3.FB & SV2.FB & SV1.FB & !SV0.FB ;

CLR\_RC3= SV5.FB & !SV4.FB & !SV3.FB & SV2.FB & SV1.FB & !SV0.FB ;

CLR\_SCAN= !SV5.FB & SV4.FB & !SV3.FB & !SV2.FB & !SV1.FB & SV0.FB ;

```
EN_STRB= SV4.FB & SV3.FB & !SV2.FB & !SV1.FB & !SV0.FB # !SV5.FB & !SV4.FB  
    & !SV3.FB & SV2.FB & !SV1.FB & !SV0.FB # SV4.FB & SV3.FB & SV2.FB & SV1.FB  
    & !SV0.FB ;
```

INC\_DIG= SV5.FB & !SV4.FB & !SV3.FB & SV2.FB & !SV1.FB & !SV0.FB ;

INC\_L2Q= !SV5.FB & SV4.FB & SV3.FB & !SV2.FB & !SV1.FB & !SV0.FB ;

JOINED= SV5.FB & SV4.FB & !SV3.FB & !SV2.FB & SV1.FB & !SV0.FB # SV5.FB & SV4.FB & !SV3.FB & SV2.FB & !SV1.FB & SV0.FB ;

```
L1_DONE= SV5.FB & !SV4.FB & SV3.FB & !SV2.FB & !SV1.FB & !SV0.FB # !SV5.FB
& SV4.FB & SV3.FB & SV2.FB & !SV1.FB & !SV0.FB # SV5.FB & SV4.FB & SV3.FB &
!SV2.FB & !SV1.FB & SV0.FB # SV5.FB & !SV4.FB & !SV3.FB & SV2.FB & !SV1.FB
& SV0.FB ;
```

LATCH\_VRB= SV4.FB & !SV3.FB & !SV2.FB & !SV1.FB & !SV0.FB # SV5.FB & !  
SV3.FB & !SV2.FB & !SV1.FB & !SV0.FB # SV5.FB & !SV4.FB & SV3.FB & !SV2.FB  
& SV1.FB & !SV0.FB # !SV4.FB & SV3.FB & !SV2.FB & !SV1.FB & SV0.FB ;

P\_FIFO\_OE= SV4.FB & SV3.FB & SV2.FB & SV1.FB & !SV0.FB # !SV5.FB & SV4.FB & !SV3.FB & !SV2.FB & !SV1.FB & SV0.FB ;

```

PIPE_RD2= SV5.FB & !SV4.FB & SV3.FB & !SV2.FB & !SV1.FB & !SV0.FB # !SV5.FB
    & SV4.FB & SV3.FB & SV2.FB & !SV1.FB & !SV0.FB # SV5.FB & SV4.FB & SV3.FB
    & !SV2.FB & !SV1.FB & SV0.FB # SV5.FB & !SV4.FB & !SV3.FB & SV2.FB & !
    SV1.FB & SV0.FB ;

```



```

SV4.FB & !SV5.FB & ([0,0,0,0]) #( !SV0.FB & SV1.FB & SV2.FB & SV3.FB
& !SV4.FB & SV5.FB & ([0,0,0,0]) #( !SV0.FB & SV1.FB & SV2.FB &
SV3.FB & SV4.FB & !SV5.FB & ([0,0,0,0]) #( !SV0.FB & SV1.FB & SV2.FB
& SV3.FB & SV4.FB & SV5.FB & ([0,0,0,0]) #( SV0.FB & !SV1.FB & !
SV2.FB & !SV3.FB & !SV4.FB & !SV5.FB & ([0,0,0,0]) #( SV0.FB & !
SV1.FB & !SV2.FB & !SV3.FB & !SV4.FB & !SV5.FB & ([0,0,0,0]) #(
SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB & SV4.FB & !SV5.FB & ([0,0,0,0])
#( SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB & SV4.FB & SV5.FB & ([0,0,0,
0]) #( SV0.FB & !SV1.FB & !SV2.FB & SV3.FB & !SV4.FB & !SV5.FB &
([0,0,0,0]) #( SV0.FB & !SV1.FB & !SV2.FB & SV3.FB & !SV4.FB & SV5.FB
& ([0,0,0,0]) #( SV0.FB & !SV1.FB & !SV2.FB & SV3.FB & SV4.FB & !
SV5.FB & ([0,0,0,0]) #( SV0.FB & !SV1.FB & !SV2.FB & SV3.FB & SV4.FB
& SV5.FB & ([0,0,0,0]) #( SV0.FB & !SV1.FB & SV2.FB & !SV3.FB & !
SV4.FB & !SV5.FB & ([0,0,0,0]) #( SV0.FB & !SV1.FB & SV2.FB & !
SV3.FB & !SV4.FB & SV5.FB & ([0,0,0,0]) #( SV0.FB & !SV1.FB & SV2.FB
& !SV3.FB & SV4.FB & !SV5.FB & ([0,0,0,0]) #( SV0.FB & !SV1.FB &
SV2.FB & !SV3.FB & SV4.FB & SV5.FB & ([0,0,0,0]) #( SV0.FB & !SV1.FB
& SV2.FB & SV3.FB & !SV4.FB & !SV5.FB & ([0,0,0,0]));

```

SET\_SCAN= !SV5.FB & !SV4.FB & !SV3.FB & !SV2.FB & !SV1.FB & SV0.FB ;

```
SUB=( !SV6.FB & !SV7.FB & !SV8.FB & ([0,0,0,0] ) ) # ( !SV6.FB & !SV7.FB
    & SV8.FB & ([0,0,0,1] ) ) # ( !SV6.FB & SV7.FB & !SV8.FB & ([0,0,1,0] )
) # ( !SV6.FB & SV7.FB & SV8.FB & ([0,1,0,0] ) ) # ( SV6.FB & !SV7.FB &
!SV8.FB & ([1,0,0,0] ) );
```

WRITE\_EF= !SV5.FB & SV4.FB & !SV3.FB & !SV2.FB & !SV1.FB & SV0.FB ;

## "Logic Equations

## EQUATIONS

```

CRT_L2R = SUB0 & SUB1 & SUB2 & SUB3 & P_L2R0 & P_L2R1 & P_L2R2 & P_L2R3 #
    !SUB0 & SUB1 & SUB2 & SUB3 & !P_L2R0 & P_L2R1 & P_L2R2 & P_L2R3 # SUB0 & !
    SUB1 & SUB2 & SUB3 & P_L2R0 & !P_L2R1 & P_L2R2 & P_L2R3 # !SUB0 & !SUB1 &
    SUB2 & SUB3 & !P_L2R0 & !P_L2R1 & P_L2R2 & P_L2R3 # SUB0 & SUB1 & !SUB2 &
    SUB3 & P_L2R0 & P_L2R1 & !P_L2R2 & P_L2R3 # !SUB0 & SUB1 & !SUB2 & SUB3 & !
    P_L2R0 & P_L2R1 & !P_L2R2 & P_L2R3 # SUB0 & !SUB1 & !SUB2 & SUB3 & P_L2R0 &
    !P_L2R1 & !P_L2R2 & P_L2R3 # !SUB0 & !SUB1 & !SUB2 & SUB3 & !P_L2R0 & !
    P_L2R1 & !P_L2R2 & P_L2R3 # SUB0 & SUB1 & SUB2 & !SUB3 & P_L2R0 & P_L2R1 &
    P_L2R2 & !P_L2R3 # !SUB0 & SUB1 & SUB2 & !SUB3 & !P_L2R0 & P_L2R1 & P_L2R2
    & !P_L2R3 # SUB0 & !SUB1 & SUB2 & !SUB3 & P_L2R0 & !P_L2R1 & P_L2R2 & !
    P_L2R3 # !SUB0 & !SUB1 & SUB2 & !SUB3 & !P_L2R0 & !P_L2R1 & P_L2R2 & !
    P_L2R3 # SUB0 & SUB1 & !SUB2 & !SUB3 & P_L2R0 & P_L2R1 & !P_L2R2 & !P_L2R3
    # !SUB0 & SUB1 & !SUB2 & !SUB3 & !P_L2R0 & P_L2R1 & !P_L2R2 & !P_L2R3 # !
    SUB0 & !SUB1 & !SUB2 & !SUB3 & P_L2R0 & !P_L2R1 & !P_L2R2 & !P_L2R3 # !SUB0
    & !SUB1 & !SUB2 & !SUB3 & !P_L2R0 & !P_L2R1 & !P_L2R2 & !P_L2R3 ;

```

DO L2R = L2R1 & RC1 # L2R2 & RC2 # L2R3 & RC3 # L2R0 & RC0;

```

ERA = !L1_PTR0 & !L1_PTR1 & !L1_PTR2 & !L1_PTR3 & !L2R0 & !L2R1 & !L2R2 & !
      L2R3 & !L1_PFE # L1_PTR0 & !L1_PTR1 & !L1_PTR2 & !L1_PTR3 & L2R0 & !L2R1 &
      !L2R2 & !L2R3 & !L1_PFE # !L1_PTR0 & L1_PTR1 & !L1_PTR2 & !L1_PTR3 & !L2R0
      & L2R1 & !L2R2 & !L2R3 & !L1_PFE # L1_PTR0 & L1_PTR1 & !L1_PTR2 & !L1_PTR3
      & L2R0 & L2R1 & !L2R2 & !L2R3 & !L1_PFE # !L1_PTR0 & !L1_PTR1 & L1_PTR2 & !
      L1_PTR3 & !L2R0 & !L2R1 & L2R2 & !L2R3 & !L1_PFE # L1_PTR0 & !L1_PTR1 &
      L1_PTR2 & !L1_PTR3 & L2R0 & !L2R1 & L2R2 & !L2R3 & !L1_PFE # !L1_PTR0 &
      L1_PTR1 & L1_PTR2 & !L1_PTR3 & !L2R0 & L2R1 & L2R2 & !L2R3 & !L1_PFE #

```

```

L1_PTR0 & L1_PTR1 & L1_PTR2 & !L1_PTR3 & L2R0 & L2R1 & L2R2 & !L2R3 & !L1_PFE
# !L1_PTR0 & !L1_PTR1 & !L1_PTR2 & L1_PTR3 & !L2R0 & !L2R1 & !L2R2 & L2R3
& !L1_PFE # L1_PTR0 & !L1_PTR1 & !L1_PTR2 & L1_PTR3 & L2R0 & !L2R1 & !L2R2
& L2R3 & !L1_PFE # !L1_PTR0 & L1_PTR1 & !L1_PTR2 & L1_PTR3 & !L2R0 & L2R1 &
!L2R2 & L2R3 & !L1_PFE # L1_PTR0 & L1_PTR1 & !L1_PTR2 & L1_PTR3 & L2R0 &
L2R1 & !L2R2 & L2R3 & !L1_PFE # !L1_PTR0 & !L1_PTR1 & L1_PTR2 & L1_PTR3 & !
L2R0 & !L2R1 & L2R2 & L2R3 & !L1_PFE # L1_PTR0 & !L1_PTR1 & L1_PTR2 &
L1_PTR3 & L2R0 & !L2R1 & L2R2 & L2R3 & !L1_PFE # !L1_PTR0 & L1_PTR1 &
L1_PTR2 & L1_PTR3 & !L2R0 & L2R1 & L2R2 & L2R3 & !L1_PFE # L1_PTR0 &
L1_PTR1 & L1_PTR2 & L1_PTR3 & L2R0 & L2R1 & L2R2 & L2R3 & !L1_PFE ;

```

L1\_MSK = [L1\_PTR3 , L1\_PTR2 , L1\_PTR1 , L1\_PTR0 ] & [RC3 , RC2 , RC1 , RC0 ]  
;  
P\_L2R = [L2R3 , L2R2 , L2R1 , L2R0 ] & [SUB3 , SUB2 , SUB1 , SUB0 ];  
END READ\_SM6

#### 4.6.6 RES\_PR

```
" D:\CDF\SRC\RSM\RES_PR.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.2
" Thu Nov 13 14:09:46 1997

" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are area optimized.

MODULE RES_PR

DECLARATIONS
"clock name
    CLK PIN;

"Input variables
    GAP PIN;
    NO_OP PIN;
    RUN PIN;
    TIMEOUT PIN;

"Output variables
    CLR_TIMER PIN ISTYPE 'com';
    ERR4 PIN ISTYPE 'com';
    FCMD0 PIN ISTYPE 'com';
    FCMD1 PIN ISTYPE 'com';
    FCMD2 PIN ISTYPE 'com';
    FCMD3 PIN ISTYPE 'com';
    FCMD4 PIN ISTYPE 'com';
    FXQTI PIN ISTYPE 'com';
    INC_TIMER PIN ISTYPE 'com';
    STEAL_BUS PIN ISTYPE 'com';

"State variables
    SV0 PIN ISTYPE 'reg';
    SV1 PIN ISTYPE 'reg';
    SV2 PIN ISTYPE 'reg';

"Vectors
DECLARATIONS
    FCMD=[  
        FCMD4,  
        FCMD3,  
        FCMD2,  
        FCMD1,  
        FCMD0  
    ];

"State Register assignment
DECLARATIONS
```

```

sreg=[ SV0,SV1,SV2];

EQUATIONS
    sreg.clk=CLK;

DECLARATIONS
    STATE0=[0, 0, 0];
    STATE1=[0, 0, 1];
    STATE2=[0, 1, 0];
    STATE3=[0, 1, 1];
    STATE4=[1, 0, 0];
    STATE5=[1, 0, 1];
    STATE6=[1, 1, 0];

EQUATIONS
    SV0 := NO_OP & GAP & !SV0.FB # SV0.FB & !SV1.FB & !SV2.FB # NO_OP & !
        SV0.FB & SV1.FB # !SV0.FB & SV1.FB & SV2.FB # NO_OP & !SV0.FB & SV2.FB #
        NO_OP & SV0.FB & !SV1.FB # GAP & SV0.FB & !SV2.FB # !RUN ;
    SV1 := !SV0.FB & SV1.FB # SV0.FB & !SV1.FB & SV2.FB # GAP & SV1.FB & !
        SV2.FB # !RUN ;
    SV2 := RUN & !NO_OP & GAP & !SV0.FB & !SV1.FB # RUN & !NO_OP & !SV0.FB & !
        SV1.FB & SV2.FB # RUN & !NO_OP & !SV0.FB & SV1.FB & !SV2.FB # RUN &
        TIMEOUT & SV0.FB & !SV1.FB & !SV2.FB ;

CLR_TIMER= !SV2.FB & !SV1.FB & !SV0.FB ;

ERR4= RUN & !NO_OP & !SV0.FB & SV1.FB & SV2.FB # !GAP & RUN & NO_OP & !
    SV0.FB & SV1.FB # !GAP & RUN & NO_OP & SV0.FB & !SV1.FB & SV2.FB ;

FCMD= ( SV0.FB & !SV1.FB & !SV2.FB & ([0,0,0,0,0] ) ) # ( SV0.FB & SV1.FB
    & !SV2.FB & ([0,0,0,0,0] ) ) # ( !SV0.FB & !SV1.FB & !SV2.FB & ( !GAP &
    RUN ) & ([0,0,0,0,0] ) ) # ( !SV0.FB & !SV1.FB & !SV2.FB & ( RUN & !
    NO_OP & GAP ) & ([0,0,0,0,0] ) ) # ( !SV0.FB & !SV1.FB & !SV2.FB & ( RUN
    & NO_OP & GAP ) & ([0,0,1,0,0] ) ) # ( !SV0.FB & !SV1.FB & SV2.FB & ( RUN
    & NO_OP ) & ([0,0,1,0,0] ) ) # ( !SV0.FB & !SV1.FB & SV2.FB & ( RUN & !
    NO_OP ) & ([0,0,0,0,0] ) ) # ( !SV0.FB & SV1.FB & !SV2.FB & ( RUN & !
    NO_OP ) & ([0,0,0,0,0] ) ) # ( !SV0.FB & SV1.FB & !SV2.FB & ( RUN & NO_OP
    ) & ([0,0,0,1,0] ) ) # ( !SV0.FB & SV1.FB & SV2.FB & ( RUN & !NO_OP ) &
    ([0,0,0,0,0] ) ) # ( !SV0.FB & SV1.FB & SV2.FB & ( RUN & NO_OP ) & ([0,0
    ,0,1,0] ) ) # ( SV0.FB & !SV1.FB & SV2.FB & ( RUN & NO_OP ) & ([0,0,0,1,
    0] ) ) # ( SV0.FB & !SV1.FB & SV2.FB & ( RUN & !NO_OP ) & ([0,0,0,0,0] )
    );

FXQTI= RUN & NO_OP & GAP & !SV0.FB # RUN & NO_OP & !SV0.FB & SV1.FB # RUN
    & NO_OP & !SV0.FB & SV2.FB # RUN & NO_OP & !SV1.FB & SV2.FB ;

INC_TIMER= SV2.FB & !SV1.FB & !SV0.FB # !SV2.FB & !SV1.FB & SV0.FB ;

STEAL_BUS= RUN & NO_OP & GAP & !SV0.FB # RUN & NO_OP & !SV0.FB & SV1.FB #
    RUN & NO_OP & !SV0.FB & SV2.FB # RUN & NO_OP & !SV1.FB & SV2.FB ;
END RES_PR

```



#### 4.6.7 RSMMODES

```
" D:\CDF\SRC\RSM\RSMMODES.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.2
" Wed Nov 26 09:55:09 1997

" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are manually optimized.
```

```
MODULE RSMMODES
```

##### DECLARATIONS

```
"clock name
    CLK PIN;
```

```
"Input variables
```

```
    CREG0 PIN;
    CREG1 PIN;
    CREG2 PIN;
    CREG3 PIN;
    CREG4 PIN;
    CREG5 PIN;
    CREG6 PIN;
    CREG7 PIN;
    HALT PIN;
    RECOVER PIN;
    RUN PIN;
    SCAN0 PIN;
    SCAN1 PIN;
    SCAN2 PIN;
    SCAN3 PIN;
    SRC2FIB PIN;
    XQT PIN;
```

```
"Output variables
```

```
    BM0 PIN ISTYPE 'com';
    BM1 PIN ISTYPE 'com';
    BM2 PIN ISTYPE 'com';
    BM3 PIN ISTYPE 'com';
    DO_IT PIN ISTYPE 'com';
    DUMP_Q PIN ISTYPE 'com';
    EFDATA PIN ISTYPE 'com';
    L1_ADD0 PIN ISTYPE 'com';
    L1_ADD1 PIN ISTYPE 'com';
    PTRDATA PIN ISTYPE 'com';
    RES_BM PIN ISTYPE 'reg';
    RES_FIFOS PIN ISTYPE 'com';
    SEL_BM PIN ISTYPE 'com';
    VME_ACCESS PIN ISTYPE 'com';
    WRT_BP PIN ISTYPE 'com';
```

```

WRT_E_FIFO PIN ISTYPE 'com';

"State variables
SV0 PIN ISTYPE 'reg';
SV1 PIN ISTYPE 'reg';
SV2 PIN ISTYPE 'reg';
SV3 PIN ISTYPE 'reg';
SV4 PIN ISTYPE 'reg';

"Vectors
DECLARATIONS
BM=[  

    BM3,  

    BM2,  

    BM1,  

    BM0  

];  

L1_ADD=[  

    L1_ADD1,  

    L1_ADD0  

];  

MODE=[  

    CREG7,  

    CREG6,  

    CREG5,  

    CREG4  

];  

SCAN=[  

    SCAN3,  

    SCAN2,  

    SCAN1,  

    SCAN0  

];  

SM=[  

    CREG3,  

    CREG2,  

    CREG1,  

    CREG0  

];  

"Clock logic clock setup
EQUATIONS
RES_BM.clk=CLK;  

"State Register assignment
DECLARATIONS
sreg=[ SV0,SV1,SV2,SV3,SV4];  

EQUATIONS
sreg.clk=CLK;

```

## DECLARATIONS

```

STATE0=[0, 0, 0, 0, 0];
STATE1=[0, 0, 0, 0, 1];
STATE2=[0, 0, 0, 1, 0];
STATE3=[0, 0, 0, 1, 1];
STATE4=[0, 0, 1, 0, 0];
STATE5=[0, 0, 1, 0, 1];
STATE6=[0, 0, 1, 1, 0];
STATE7=[0, 0, 1, 1, 1];
STATE8=[0, 1, 0, 0, 0];
STATE9=[0, 1, 0, 0, 1];
STATE10=[0, 1, 0, 1, 0];
STATE11=[0, 1, 0, 1, 1];
STATE12=[0, 1, 1, 0, 0];
STATE13=[0, 1, 1, 0, 1];
STATE14=[0, 1, 1, 1, 0];
STATE15=[0, 1, 1, 1, 1];
STATE16=[1, 0, 0, 0, 0];
STATE17=[1, 0, 0, 0, 1];
STATE18=[1, 0, 0, 1, 0];
STATE20=[1, 0, 0, 1, 1];
STATE21=[1, 0, 1, 0, 0];
STATE22=[1, 0, 1, 0, 1];

```

## EQUATIONS

```

SV0 := SCAN3 & !SV0.FB & SV1.FB & SV2.FB & SV3.FB & SV4.FB # !SCAN2 & !
SV0.FB & SV1.FB & SV2.FB & SV3.FB & SV4.FB # SCAN1 & !SV0.FB & SV1.FB &
SV2.FB & SV3.FB & SV4.FB # !SCAN0 & !SV0.FB & SV1.FB & SV2.FB & SV3.FB &
SV4.FB # SCAN0 & SV0.FB & !SV1.FB & !SV3.FB & !SV4.FB # SCAN3 & SV0.FB &
!SV1.FB & !SV2.FB & !SV3.FB # !SCAN2 & SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB
# !SCAN1 & SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB # !SCAN0 & SV0.FB & !
SV1.FB & !SV2.FB & !SV3.FB & SV4.FB # !CREG3 & !CREG2 & CREG1 & CREG0 & XQT
& !CREG7 & !CREG6 & !CREG5 & !CREG4 & !SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB
& !SV4.FB # SV0.FB & !SV1.FB & SV2.FB & !SV3.FB & !SV4.FB # CREG5 &
RECOVER & SV0.FB & !SV1.FB & SV2.FB & !SV3.FB # !CREG7 & !CREG6 & CREG5 &
CREG4 & HALT & !SV0.FB & !SV1.FB & !SV2.FB & SV3.FB & SV4.FB # !XQT & HALT
& !RECOVER & SV0.FB & !SV1.FB & SV2.FB & !SV3.FB # CREG5 & HALT & SV0.FB &
!SV1.FB & SV2.FB & !SV3.FB # CREG4 & HALT & !RECOVER & SV0.FB & !SV1.FB &
SV2.FB & !SV3.FB # !XQT & !CREG4 & SV0.FB & !SV1.FB & SV2.FB & !SV3.FB #
CREG5 & !CREG4 & SV0.FB & !SV1.FB & SV2.FB & !SV3.FB # !XQT & !CREG5 &
SV0.FB & !SV1.FB & SV2.FB & !SV3.FB # CREG4 & !CREG5 & SV0.FB & !SV1.FB &
SV2.FB & !SV3.FB # CREG6 & SV0.FB & !SV1.FB & SV2.FB & !SV3.FB # CREG7 &
SV0.FB & !SV1.FB & SV2.FB & !SV3.FB ;

SV1 := !SV0.FB & !SV1.FB & SV2.FB & SV3.FB & SV4.FB # !SCAN3 & SCAN2 & !
SCAN1 & SCAN0 & !SV0.FB & SV2.FB & SV3.FB & SV4.FB # !SCAN3 & SCAN2 &
SCAN1
& !SCAN0 & SV0.FB & !SV1.FB & !SV2.FB & !SV4.FB # !SCAN3 & SCAN2 & SCAN1 &
SCAN0 & SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB & SV4.FB # SV0.FB & !SV1.FB &
!SV2.FB & SV3.FB & !SV4.FB # !SV0.FB & SV1.FB & !SV2.FB & SV4.FB # !
SV0.FB & SV1.FB & !SV3.FB # !SV0.FB & SV1.FB & SV2.FB & !SV4.FB ;

SV2 := RUN & CREG4 & !SV0.FB & !SV1.FB & SV2.FB & !SV4.FB # RUN & CREG5 & !

```





SV3.FB ;

DO\_IT= !SV4.FB & !SV3.FB & SV2.FB & !SV1.FB & !SV0.FB ;

DUMP\_Q= !SV4.FB & SV3.FB & !SV2.FB & !SV1.FB & !SV0.FB # !SV4.FB & !SV3.FB  
  & SV2.FB & !SV1.FB & SV0.FB ;

EFDATA= !SV3.FB & SV2.FB & SV1.FB & !SV0.FB # !SV4.FB & SV2.FB & SV1.FB & !  
SV0.FB # SV4.FB & SV3.FB & SV1.FB & !SV0.FB # !SV3.FB & !SV2.FB & !SV1.FB  
& SV0.FB # !SV4.FB & !SV2.FB & !SV1.FB & SV0.FB ;

```

L1_ADD=( !SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB & !SV4.FB & ([0,0]) )#
( !SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB & SV4.FB & ([0,0]) )#( !
SV0.FB & !SV1.FB & !SV2.FB & SV3.FB & !SV4.FB & ([0,0]) )#( !SV0.FB
& !SV1.FB & !SV2.FB & SV3.FB & SV4.FB & ([0,0]) )#( !SV0.FB & !SV1.FB
& SV2.FB & !SV3.FB & !SV4.FB & ([0,0]) )#( !SV0.FB & !SV1.FB &
SV2.FB & !SV3.FB & SV4.FB & ([0,0]) )#( !SV0.FB & !SV1.FB & SV2.FB &
SV3.FB & !SV4.FB & ([0,0]) )#( !SV0.FB & !SV1.FB & SV2.FB & SV3.FB
& SV4.FB & ([0,1]) )#( !SV0.FB & SV1.FB & !SV2.FB & !SV3.FB & !SV4.FB
& ([1,0]) )#( !SV0.FB & SV1.FB & !SV2.FB & !SV3.FB & SV4.FB & ([1,
1]) )#( !SV0.FB & SV1.FB & !SV2.FB & SV3.FB & !SV4.FB & ([0,0]) )#
( !SV0.FB & SV1.FB & !SV2.FB & SV3.FB & SV4.FB & ([0,0]) )#( !SV0.FB
& SV1.FB & SV2.FB & !SV3.FB & !SV4.FB & ([0,0]) )#( !SV0.FB & SV1.FB
& SV2.FB & !SV3.FB & SV4.FB & ([0,0]) )#( !SV0.FB & SV1.FB & SV2.FB
& SV3.FB & !SV4.FB & ([0,0]) )#( !SV0.FB & SV1.FB & SV2.FB & SV3.FB
& SV4.FB & ([0,0]) )#( SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB & !
SV4.FB & ([0,0]) )#( SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB & SV4.FB &
([0,0]) )#( SV0.FB & !SV1.FB & !SV2.FB & SV3.FB & !SV4.FB & ([0,0]) )
#( SV0.FB & !SV1.FB & !SV2.FB & SV3.FB & SV4.FB & ([0,0]) )#( SV0.FB &
!SV1.FB & !SV2.FB & !SV3.FB & !SV4.FB & ([0,0]) )#( SV0.FB & !SV1.FB
& SV2.FB & !SV3.FB & SV4.FB & ([0,0]) );

```

```

PTRDATA= SV3.FB & SV2.FB & !SV1.FB & !SV0.FB # !SV3.FB & !SV2.FB & SV1.FB &
!SV0.FB ;

RES_BM := XQT & !CREG3 & !CREG2 & CREG1 & !CREG0 & !CREG7 & !CREG6 & !CREG5 &
!CREG4 & !SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB & !SV4.FB # !CREG7 & !CREG6
& CREG5 & CREG4 & RECOVER & SV0.FB & !SV1.FB & SV2.FB & !SV3.FB & SV4.FB ;

RES_FIFOS= SV4.FB & !SV3.FB & !SV2.FB & !SV1.FB & !SV0.FB # !SV4.FB & !
SV3.FB & SV2.FB & !SV1.FB & SV0.FB ;

SEL_BM= SV4.FB & SV3.FB & !SV2.FB & !SV1.FB & !SV0.FB # !SV4.FB & !SV3.FB &
!SV1.FB & !SV0.FB # !SV3.FB & SV2.FB & !SV1.FB & !SV0.FB # SV4.FB & !
SV3.FB & SV2.FB & !SV1.FB ;

VME_ACCESS= !SV4.FB & !SV3.FB & !SV2.FB & !SV1.FB & !SV0.FB ;

WRT_BP= SV3.FB & SV2.FB & !SV1.FB & !SV0.FB # !SV3.FB & !SV2.FB & SV1.FB &
!SV0.FB ;

WRT_E_FIFO= !SV3.FB & SV2.FB & SV1.FB & !SV0.FB # !SV4.FB & SV2.FB & SV1.FB
& !SV0.FB # SV4.FB & SV3.FB & SV1.FB & !SV0.FB # !SV3.FB & !SV2.FB & !
SV1.FB & SV0.FB # !SV4.FB & !SV2.FB & !SV1.FB & SV0.FB ;
END RSMMODES

```

#### 4.6.8 SCAN\_Q

```
" D:\CDF\SRC\RSM\SCAN_Q.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.2
" Thu Nov 13 15:29:32 1997
```

```
" This Abel code was generated using:
" binary encoded state assignment with structured code format.
" Minimization is enabled, implied else is enabled,
" and outputs are area optimized.
```

```
MODULE SCAN_Q
```

```
DECLARATIONS
```

```
"clock name
    CLK PIN;
```

```
"Input variables
```

```
    DN PIN;
    DUMP_Q PIN;
    UP PIN;
```

```
"Output variables
```

```
    SCANQ0 PIN ISTYPE 'com';
    SCANQ1 PIN ISTYPE 'com';
    SCANQ2 PIN ISTYPE 'com';
    SCANQ3 PIN ISTYPE 'com';
    SCANQ4 PIN ISTYPE 'com';
    SCANQ5 PIN ISTYPE 'com';
    SCANQ6 PIN ISTYPE 'com';
    SCANQ7 PIN ISTYPE 'com';
```

```
"State variables
```

```
    SV0 PIN ISTYPE 'reg';
    SV1 PIN ISTYPE 'reg';
    SV2 PIN ISTYPE 'reg';
    SV3 PIN ISTYPE 'reg';
```

```
"Vectors
```

```
DECLARATIONS
```

```
    SCANQ=[
        SCANQ7,
        SCANQ6,
        SCANQ5,
        SCANQ4,
        SCANQ3,
        SCANQ2,
        SCANQ1,
        SCANQ0
    ];
```

```
"State Register assignment
```

```

DECLARATIONS
    sreg=[ SV0,SV1,SV2,SV3];

EQUATIONS
    sreg.clk=CLK;

DECLARATIONS
    STATE0=[0, 0, 0, 0];
    STATE1=[0, 0, 0, 1];
    STATE2=[0, 0, 1, 0];
    STATE3=[0, 0, 1, 1];
    STATE4=[0, 1, 0, 0];
    STATE5=[0, 1, 0, 1];
    STATE6=[0, 1, 1, 0];
    STATE7=[0, 1, 1, 1];
    STATE8=[1, 0, 0, 0];

" Asynchronous Reset
EQUATIONS
    [SV0, SV1, SV2, SV3].ar = DUMP_Q ;

state_diagram sreg;

state STATE0:
    SCANQ=[1,1,1,1,1,1,1];
    IF ( UP & !DN ) THEN STATE1;
    IF ( DN # !UP ) THEN STATE0;
state STATE1:
    SCANQ=[0,1,1,1,1,1,1];
    IF ( UP & !DN ) THEN STATE2;
    IF ( !UP & !DN # DN & UP ) THEN STATE1;
    IF ( DN & !UP ) THEN STATE0;
state STATE2:
    SCANQ=[0,0,1,1,1,1,1];
    IF ( UP & !DN ) THEN STATE3;
    IF ( DN & !UP ) THEN STATE1;
    IF ( !UP & !DN # DN & UP ) THEN STATE2;
state STATE3:
    SCANQ=[0,0,0,1,1,1,1];
    IF ( UP & !DN ) THEN STATE4;
    IF ( DN & !UP ) THEN STATE2;
    IF ( !UP & !DN # DN & UP ) THEN STATE3;
state STATE4:
    SCANQ=[0,0,0,0,1,1,1];
    IF ( !UP & !DN # DN & UP ) THEN STATE4;
    IF ( UP & !DN ) THEN STATE5;
    IF ( DN & !UP ) THEN STATE3;
state STATE5:
    SCANQ=[0,0,0,0,0,1,1];
    IF ( !UP & !DN # DN & UP ) THEN STATE5;
    IF ( UP & !DN ) THEN STATE6;
    IF ( DN & !UP ) THEN STATE4;
state STATE6:
    SCANQ=[0,0,0,0,0,1,1];

```

```
IF ( !UP & !DN # DN & UP ) THEN STATE6;
IF ( UP & !DN ) THEN STATE7;
IF ( DN & !UP ) THEN STATE5;
state STATE7:
SCANQ=[0,0,0,0,0,0,1];
IF ( !UP & !DN # DN & UP ) THEN STATE7;
IF ( UP & !DN ) THEN STATE8;
IF ( DN & !UP ) THEN STATE6;
state STATE8:
SCANQ=[0,0,0,0,0,0,0];
IF ( DN & !UP ) THEN STATE7;
IF ( UP # !DN ) THEN STATE8;
END SCAN_Q
```

#### 4.6.9 SRC\_ERR1

```
" D:\CDF\SRC\RSM\SRC_ERR1.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.2
" Thu Nov 13 15:30:05 1997

" This Abel code was generated using:
" binary encoded state assignment with structured code format.
" Minimization is enabled, implied else is enabled,
" and outputs are area optimized.
```

```
MODULE SRC_ERR1
```

```
DECLARATIONS
```

```
"clock name
    CLK PIN;
```

```
"Input variables
```

```
    DUMP_Q PIN;
    L1_DONE PIN;
    L1A PIN;
```

```
"Output variables
```

```
    L1Q0 PIN ISTYPE 'com';
    L1Q1 PIN ISTYPE 'com';
    L1Q2 PIN ISTYPE 'com';
    L1Q3 PIN ISTYPE 'com';
    SRC_ERR1 PIN ISTYPE 'com';
```

```
"State variables
```

```
    SV0 PIN ISTYPE 'reg';
    SV1 PIN ISTYPE 'reg';
    SV2 PIN ISTYPE 'reg';
```

```
"Vectors
```

```
DECLARATIONS
```

```
    L1Q=[
        L1Q3,
        L1Q2,
        L1Q1,
        L1Q0
    ];
```

```
"State Register assignment
```

```
DECLARATIONS
```

```
    sreg=[ SV0,SV1,SV2];
```

```
EQUATIONS
```

```
    sreg.clk=CLK;
```

```
DECLARATIONS
```

```
    STATE0=[0, 0, 0];
```

```

STATE2=[0, 0, 1];
STATE3=[0, 1, 0];
STATE4=[0, 1, 1];
STATE5=[1, 0, 0];
STATE6=[1, 0, 1];

EQUATIONS
WHEN (!((
    (sreg.FB==STATE0)
    # (sreg.FB==STATE2)
    # (sreg.FB==STATE3)
    # (sreg.FB==STATE4)
    # (sreg.FB==STATE5)
    # (sreg.FB==STATE6)
) &
    ( DUMP_Q )) THEN {
    [sreg] := [STATE0];
}

WHEN (!((
    (sreg.FB==STATE0)
    # (sreg.FB==STATE2)
    # (sreg.FB==STATE3)
    # (sreg.FB==STATE4)
    # (sreg.FB==STATE5)
    # (sreg.FB==STATE6)
) &
    !( DUMP_Q )) THEN {
    [sreg] := [STATE0];
}

state_diagram sreg;

state STATE0:
SRC_ERR1=0;
L1Q=[1,1,1,1];
IF ( DUMP_Q ) THEN STATE0;
ELSE
    IF ( L1A ) THEN STATE2;
    IF ( !L1A ) THEN STATE0;

state STATE2:
SRC_ERR1=0;
L1Q=[0,1,1,1];
IF ( DUMP_Q ) THEN STATE0;
ELSE
    IF ( L1A & L1_DONE ) THEN STATE2;
    IF ( !L1_DONE & L1A ) THEN STATE3;
    IF ( !L1A & L1_DONE ) THEN STATE0;
    IF ( !L1_DONE & !L1A ) THEN STATE2;

state STATE3:
SRC_ERR1=0;
L1Q=[0,0,1,1];
IF ( DUMP_Q ) THEN STATE0;
ELSE
    IF ( L1A & L1_DONE ) THEN STATE3;

```

```

        IF ( !L1_DONE & L1A ) THEN STATE4;
        IF ( !L1A & L1_DONE ) THEN STATE2;
        IF ( !L1_DONE & !L1A ) THEN STATE3;
state STATE4:
        SRC_ERR1=0;
        L1Q=[0,0,0,1];
        IF ( DUMP_Q ) THEN STATE0;
        ELSE
            IF ( L1A & L1_DONE ) THEN STATE4;
            IF ( !L1_DONE & L1A ) THEN STATE5;
            IF ( !L1A & L1_DONE ) THEN STATE3;
            IF ( !L1_DONE & !L1A ) THEN STATE4;
state STATE5:
        SRC_ERR1=0;
        L1Q=[0,0,0,0];
        IF ( DUMP_Q ) THEN STATE0;
        ELSE
            IF ( L1A & L1_DONE ) THEN STATE5;
            IF ( !L1_DONE & L1A ) THEN STATE6;
            IF ( !L1A & L1_DONE ) THEN STATE4;
            IF ( !L1_DONE & !L1A ) THEN STATE5;
state STATE6:
        SRC_ERR1=1;
        L1Q=[0,0,0,0];
        IF ( DUMP_Q ) THEN STATE0;
        ELSE
            IF ( !0 ) THEN STATE5;
END SRC_ERR1

```

#### 4.6.10 SRC\_ERR2

```
" D:\CDF\SRC\RSM\SRC_ERR2.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.2
" Thu Nov 13 15:32:29 1997

" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are area optimized.
```

```
MODULE SRC_ERR2
```

##### DECLARATIONS

```
"clock name
    CLK PIN;
```

##### "Input variables

```
DUMP_Q PIN;
L1A PIN;
L1PTR0 PIN;
L1PTR1 PIN;
L1PTR2 PIN;
L1PTR3 PIN;
L2A PIN;
L2A0 PIN;
L2A1 PIN;
L2A2 PIN;
L2A3 PIN;
L2R PIN;
L2R0 PIN;
L2R1 PIN;
L2R2 PIN;
L2R3 PIN;
```

##### "Output variables

```
ERR20 PIN ISTYPE 'com';
ERR21 PIN ISTYPE 'com';
ERR22 PIN ISTYPE 'com';
ERR23 PIN ISTYPE 'com';
ERR30 PIN ISTYPE 'com';
ERR31 PIN ISTYPE 'com';
ERR32 PIN ISTYPE 'com';
ERR33 PIN ISTYPE 'com';
```

##### "State variables

```
SV0 PIN ISTYPE 'reg';
SV1 PIN ISTYPE 'reg';
SV2 PIN ISTYPE 'reg';
SV3 PIN ISTYPE 'reg';
SV4 PIN ISTYPE 'reg';
SV5 PIN ISTYPE 'reg';
```

```

SV6 PIN ISTYPE 'reg';
SV7 PIN ISTYPE 'reg';

"State Register assignment
DECLARATIONS
    sreg=[ SV0,SV1];

EQUATIONS
    sreg.clk=CLK;

DECLARATIONS
    STATE0=[0, 0];
    STATE1=[0, 1];
    STATE2=[1, 0];
    STATE6=[1, 1];

EQUATIONS

SV0 := !DUMP_Q & L1A & L1PTR0 & !SV0.FB & SV1.FB # !DUMP_Q & L2R & L2R0 & !
SV0.FB & !SV1.FB # !DUMP_Q & L2A & L2A0 & !SV0.FB & !SV1.FB ;

SV1 := !DUMP_Q & !L2A0 & !L2R0 & L1A & L1PTR0 & !SV1.FB # !DUMP_Q & !L2A &
!L2R0 & L1A & L1PTR0 & !SV1.FB # !DUMP_Q & !L2A0 & !L2R & L1A & L1PTR0 & !
SV1.FB # !DUMP_Q & !L2A & !L2R & L1A & L1PTR0 & !SV1.FB # !L1A & !L2R0 &
!L2A0 & !DUMP_Q & !SV0.FB & SV1.FB # !L1A & !L2R & !L2A0 & !DUMP_Q & !
SV0.FB & SV1.FB # !L1A & !L2R0 & !L2A & !DUMP_Q & !SV0.FB & SV1.FB # !L1A
& !L2R & !L2A & !DUMP_Q & !SV0.FB & SV1.FB # !L1PTR0 & !L2R0 & !L2A0 & !
DUMP_Q & !SV0.FB & SV1.FB # !L1PTR0 & !L2R & !L2A0 & !DUMP_Q & !SV0.FB &
SV1.FB # !L1PTR0 & !L2R0 & !L2A & !DUMP_Q & !SV0.FB & SV1.FB # !L1PTR0 &
!L2R & !L2A & !DUMP_Q & !SV0.FB & SV1.FB # !DUMP_Q & !SV0.FB & !SV1.FB # !
DUMP_Q & L2R & L2R0 & !SV1.FB # !DUMP_Q & L2A & L2A0 & !SV1.FB ;

"State Register assignment
DECLARATIONS
    sreg1=[ SV2,SV3];

EQUATIONS
    sreg1.clk=CLK;

DECLARATIONS
    STATE3=[0, 0];
    STATE4=[0, 1];
    STATE5=[1, 0];
    STATE7=[1, 1];

EQUATIONS

SV2 := !DUMP_Q & L1A & L1PTR1 & !SV2.FB & SV3.FB # !DUMP_Q & L2R & L2R0 & !
SV2.FB & !SV3.FB # !DUMP_Q & L2A & L2A0 & !SV2.FB & !SV3.FB ;

SV3 := !DUMP_Q & !L2A0 & !L2R0 & L1A & L1PTR1 & !SV3.FB # !DUMP_Q & !L2A &
!L2R0 & L1A & L1PTR1 & !SV3.FB # !DUMP_Q & !L2A0 & !L2R & L1A & L1PTR1 & !
SV3.FB # !DUMP_Q & !L2A & !L2R & L1A & L1PTR1 & !SV3.FB # !L1A & !L2R1 &
!L2A1 & !DUMP_Q & !SV2.FB & SV3.FB # !L1A & !L2R & !L2A1 & !DUMP_Q & !
SV2.FB & SV3.FB # !L1A & !L2R1 & !L2A & !DUMP_Q & !SV2.FB & SV3.FB # !L1A

```

```

& !L2R & !L2A & !DUMP_Q & !SV2.FB & SV3.FB # !L1PTR1 & !L2R1 & !L2A1 & !
DUMP_Q & !SV2.FB & SV3.FB # !L1PTR1 & !L2R & !L2A1 & !DUMP_Q & !SV2.FB &
SV3.FB # !L1PTR1 & !L2R1 & !L2A & !DUMP_Q & !SV2.FB & SV3.FB # !L1PTR1 &
!L2R & !L2A & !DUMP_Q & !SV2.FB & SV3.FB # !DUMP_Q & SV2.FB & !SV3.FB # !
DUMP_Q & L2R & L2R0 & !SV3.FB # !DUMP_Q & L2A & L2A0 & !SV3.FB ;

```

"State Register assignment

DECLARATIONS

```
sreg2=[ SV4,SV5];
```

EQUATIONS

```
sreg2.clk=CLK;
```

DECLARATIONS

```
STATE8=[0, 0];
STATE10=[0, 1];
STATE11=[1, 0];
STATE12=[1, 1];
```

EQUATIONS

```
SV4 := !L2A3 & !L2R3 & SV5.FB # !L2A & !L2R3 & SV5.FB # !L2A3 & !L2R &
SV5.FB # !L2A & !L2R & SV5.FB # !SV4.FB # !L1PTR3 & L2R & L2R3 & !
SV5.FB # !L1A & L2R & L2R3 & !SV5.FB # !L1PTR3 & L2A & L2A3 & !SV5.FB #
!L1A & L2A & L2A3 & !SV5.FB # !L2R3 & !L2A & !L1PTR3 # !L2R & !L2A & !
L1PTR3 # !L2R3 & !L2A & !L1A # !L2R & !L2A & !L1A # !L2R3 & !L2A3 & !
L1PTR3 # !L2R3 & !L2A3 & !L1A # !L2R & !L2A3 & !L1PTR3 # !L2R & !L2A3 &
!L1A # DUMP_Q ;
```

```
SV5 := L1A & L1PTR3 & !SV5.FB # !SV4.FB & !SV5.FB # !L1PTR3 & L2R & L2R3
& !SV5.FB # !L1A & L2R & L2R3 & !SV5.FB # !L1PTR3 & L2A & L2A3 & !SV5.FB
# !L1A & L2A & L2A3 & !SV5.FB # !L2R3 & !L2A & !L1PTR3 & SV4.FB & SV5.FB
# !L2R & !L2A & !L1PTR3 & SV4.FB & SV5.FB # !L2R3 & !L2A & !L1A & SV4.FB
& SV5.FB # !L2R & !L2A & !L1A & SV4.FB & SV5.FB # !L2R3 & !L2A3 & !L1PTR3
& SV4.FB & SV5.FB # !L2R3 & !L2A3 & !L1A & SV4.FB & SV5.FB # !L2R & !
L2A3 & !L1PTR3 & SV4.FB & SV5.FB # !L2R & !L2A3 & !L1A & SV4.FB & SV5.FB #
DUMP_Q ;
```

"State Register assignment

DECLARATIONS

```
sreg3=[ SV6,SV7];
```

EQUATIONS

```
sreg3.clk=CLK;
```

DECLARATIONS

```
STATE9=[0, 0];
STATE13=[0, 1];
STATE14=[1, 0];
STATE15=[1, 1];
```

EQUATIONS

```
SV6 := !L2A2 & !L2R2 & SV7.FB # !L2A & !L2R2 & SV7.FB # !L2A2 & !L2R &
SV7.FB # !L2A & !L2R & SV7.FB # !SV6.FB # !L1PTR2 & L2R & L2R2 & !
```

```

SV7.FB # !L1A & L2R & L2R2 & !SV7.FB # !L1PTR2 & L2A & L2A2 & !SV7.FB #
!L1A & L2A & L2A2 & !SV7.FB # !L2R2 & !L2A & !L1PTR2 # !L2R & !L2A & !
L1PTR2 # !L2R2 & !L2A & !L1A # !L2R & !L2A & !L1A # !L2R2 & !L2A2 & !
L1PTR2 # !L2R2 & !L2A2 & !L1A # !L2R & !L2A2 & !L1PTR2 # !L2R & !L2A2 &
!L1A # DUMP_Q ;

SV7 := L1A & L1PTR2 & !SV7.FB # !SV6.FB & !SV7.FB # !L1PTR2 & L2R & L2R2
& !SV7.FB # !L1A & L2R & L2R2 & !SV7.FB # !L1PTR2 & L2A & L2A2 & !SV7.FB
# !L1A & L2A & L2A2 & !SV7.FB # !L2R2 & !L2A & !L1PTR2 & SV6.FB & SV7.FB
# !L2R & !L2A & !L1PTR2 & SV6.FB & SV7.FB # !L2R2 & !L2A & !L1A & SV6.FB
& SV7.FB # !L2R & !L2A & !L1A & SV6.FB & SV7.FB # !L2R2 & !L2A2 & !L1PTR2
& SV6.FB & SV7.FB # !L2R2 & !L2A2 & !L1A & SV6.FB & SV7.FB # !L2R & !
L2A2 & !L1PTR2 & SV6.FB & SV7.FB # !L2R & !L2A2 & !L1A & SV6.FB & SV7.FB #
DUMP_Q ;

ERR20= SV1.FB & SV0.FB ;

ERR21= SV3.FB & SV2.FB ;

ERR22= !SV7.FB & !SV6.FB ;

ERR23= !SV5.FB & !SV4.FB ;

ERR30= !SV1.FB & SV0.FB ;

ERR31= !SV3.FB & SV2.FB ;

ERR32= SV7.FB & !SV6.FB ;

ERR33= SV5.FB & !SV4.FB ;
END SRC_ERR2

```



## 5. Buffer Manager

### 5.1 Pin Description

Design				Buffer Manager		
Date				24-Oct-96		
				3/20/98 2:16 PM		
Signal	I	O	OT	I/O	Function	Source/Destination
					<b>VRB Interface</b>	
VRB_DAT[3:0]				4	VRB command bus data lines	VRB buffer fifos
/VRB_DAT_OE	1				VRB data output enable	RSM
	5					
					<b>Miscellaneous</b>	
RSM_CLK	1				132ns state machine clock	Mastclk
/ADDR[2:1]	2				Address for VME access to pointers	RSM
	3					
					<b>Pointers</b>	
/L1A_ADD[1:0]	2				L1 Accept pointer	RSM
/WRT_BP	1				Write Buffer Pointer	RSM
/L2A_ADD[1:0]	2				L2 Accept address	RSM
/L1_PFE		1			L1 Pointer fifo Empty	RSM
/L1_PFF		1			L1 Pointer fifo Full	RSM
/WRT_L1PF	1				Write L1 Pointer fifo	RSM
/RD_L1PF	1				Read L1 Pointer fifo	RSM
/L1_PTR[1:0]		2			L1 pointer output to RSM	RSM
READADD[3:0]		4			VRB READ Address	VRB buffer fifos
	15					
					<b>Pending Register &amp; Fifo</b>	
/SEL_BUF	1				Select between SCAN address and EID	RSM
/P_FIFO_FULL		1			Pending fifo Full	RSM
/P_FIFO_EMPTY		1			Pending fifo Empty	RSM
/WRT_P_REG	1				Write Pending Register	RSM
/WRT_P_FIFO	1				Write Pending Fifo	RSM
/RD_P_FIFO	1				Read Pending Fifo	RSM
/P_FIFO_OE	1				Pending Fifo Output Enable	RSM
/E_FIFO_OE	1				Empty Fifo Output Enable	RSM
/E_FIFO_EMPTY		1			Empty fifo empty	RSM
/E_FIFO_FULL		1			Empty fifo full	RSM
/RD_E_FIFO	1				Read Empty fifo	RSM
/WRT_E_FIFO	1				Write Empty fifo	RSM
EVNTID[3:0]		4			Event ID input	Taxi
	16					
					<b>Buffer Manager Access</b>	
BM_DAT[3:0]				4	Buffer manager data io bus	RSM
/RD_BM	1				Buffer manager bus direction	RSM
/EF_DATA	1				Write Empty fifo data	RSM
/PTR_DATA	1				Write Pointer data	RSM
/PEND_2_VME	1				Read Pending fifo data to VME	RSM
/EMPTY_2_VME	1				Read Empty fifo data to VME	RSM
/PTR_2_VME	1				Read pointer data to VME	RSM
	10					
<b>Total Pin Count</b>	<b>29</b>	<b>12</b>	<b>4</b>	<b>4</b>	<b>49</b>	
						49

## 5.2 Constraint File

```
# Design: Buf_mng
# Created by XACT Floorplanner ver 6.0.1
# Pinlocked by copying in the pin file
NOTPLACE BLOCK *: P72 P33 P41;
PLACE BLOCK PTR3/$1N26: MF;
PLACE BLOCK PTR3/$1N12: NF;
PLACE BLOCK PTR3/$1I43: TBUF.LG.2;
PLACE BLOCK PTR3/$1I42: TBUF.MG.1;
PLACE BLOCK PTR3/$1I41: TBUF.NG.2;
PLACE BLOCK PTR3/$1I40: TBUF.OG.1;
PLACE BLOCK PTR3/$1I24: TBUF.MG.2;
PLACE BLOCK PTR3/$1I17: TBUF.NG.1;
PLACE BLOCK PTR3/$1I10: TBUF.OG.2;
PLACE BLOCK PTR3/$1I2: TBUF.PG.1;
PLACE BLOCK PTR2/$1N26: MG;
PLACE BLOCK PTR2/$1N12: NG;
PLACE BLOCK PTR2/$1I43: TBUF.LH.2;
PLACE BLOCK PTR2/$1I42: TBUF.MH.1;
PLACE BLOCK PTR2/$1I41: TBUF.NH.2;
PLACE BLOCK PTR2/$1I40: TBUF.OH.1;
PLACE BLOCK PTR2/$1I24: TBUF.MH.2;
PLACE BLOCK PTR2/$1I17: TBUF.NH.1;
PLACE BLOCK PTR2/$1I10: TBUF.OH.2;
PLACE BLOCK PTR2/$1I2: TBUF.PH.1;
PLACE BLOCK PTR1/$1N26: MH;
PLACE BLOCK PTR1/$1N12: NH;
PLACE BLOCK PTR1/$1I43: TBUF.LL.2;
PLACE BLOCK PTR1/$1I42: TBUF.MI.1;
PLACE BLOCK PTR1/$1I41: TBUF.NL.2;
PLACE BLOCK PTR1/$1I40: TBUF.OI.1;
PLACE BLOCK PTR1/$1I24: TBUF.MI.2;
PLACE BLOCK PTR1/$1I17: TBUF.NL.1;
PLACE BLOCK PTR1/$1I10: TBUF.OI.2;
PLACE BLOCK PTR1/$1I2: TBUF.PI.1;
PLACE BLOCK PTR0/$1N26: MI;
PLACE BLOCK PTR0/$1N12: NI;
PLACE BLOCK PTR0/$1I43: TBUF.LJ.2;
PLACE BLOCK PTR0/$1I42: TBUF.MJ.1;
PLACE BLOCK PTR0/$1I41: TBUF.NJ.2;
PLACE BLOCK PTR0/$1I40: TBUF.OJ.1;
PLACE BLOCK PTR0/$1I24: TBUF.MJ.2;
PLACE BLOCK PTR0/$1I17: TBUF.NJ.1;
PLACE BLOCK PTR0/$1I10: TBUF.OJ.2;
PLACE BLOCK PTR0/$1I2: TBUF.PJ.1;
PLACE BLOCK PEND_FIFO/REG7/R8BX7: ED;
PLACE BLOCK PEND_FIFO/REG7/R8BX5: FD;
PLACE BLOCK PEND_FIFO/REG7/R8BX3: GD;
PLACE BLOCK PEND_FIFO/REG7/R8BX1: HD;
PLACE BLOCK PEND_FIFO/REG7/$1I24: TBUF.EE.2;
PLACE BLOCK PEND_FIFO/REG7/$1I17: TBUF.FE.1;
PLACE BLOCK PEND_FIFO/REG7/$1I10: TBUF.FE.2;
PLACE BLOCK PEND_FIFO/REG7/$1I2: TBUF.GE.1;
PLACE BLOCK PEND_FIFO/REG7/$1I65: TBUF.GE.2;
```

PLACE BLOCK PEND\_FIFO/REG7/\$1I67: TBUF.HE.1;  
PLACE BLOCK PEND\_FIFO/REG7/\$1I69: TBUF.HE.2;  
PLACE BLOCK PEND\_FIFO/REG7/\$1I72: TBUF.IE.1;  
PLACE BLOCK PEND\_FIFO/REG6/R8BX7: EE;  
PLACE BLOCK PEND\_FIFO/REG6/R8BX5: FE;  
PLACE BLOCK PEND\_FIFO/REG6/R8BX3: GE;  
PLACE BLOCK PEND\_FIFO/REG6/R8BX1: HE;  
PLACE BLOCK PEND\_FIFO/REG6/\$1I24: TBUF.EF.2;  
PLACE BLOCK PEND\_FIFO/REG6/\$1I17: TBUF.FF.1;  
PLACE BLOCK PEND\_FIFO/REG6/\$1I10: TBUF.FF.2;  
PLACE BLOCK PEND\_FIFO/REG6/\$1I2: TBUF.GF.1;  
PLACE BLOCK PEND\_FIFO/REG6/\$1I65: TBUF.GF.2;  
PLACE BLOCK PEND\_FIFO/REG6/\$1I67: TBUF.HF.1;  
PLACE BLOCK PEND\_FIFO/REG6/\$1I69: TBUF.HF.2;  
PLACE BLOCK PEND\_FIFO/REG6/\$1I72: TBUF.IF.1;  
PLACE BLOCK PEND\_FIFO/REG5/R8BX7: EF;  
PLACE BLOCK PEND\_FIFO/REG5/R8BX5: FF;  
PLACE BLOCK PEND\_FIFO/REG5/R8BX3: GF;  
PLACE BLOCK PEND\_FIFO/REG5/R8BX1: HF;  
PLACE BLOCK PEND\_FIFO/REG5/\$1I24: TBUF.EG.2;  
PLACE BLOCK PEND\_FIFO/REG5/\$1I17: TBUF.FG.1;  
PLACE BLOCK PEND\_FIFO/REG5/\$1I10: TBUF.FG.2;  
PLACE BLOCK PEND\_FIFO/REG5/\$1I2: TBUF.GG.1;  
PLACE BLOCK PEND\_FIFO/REG5/\$1I65: TBUF.GG.2;  
PLACE BLOCK PEND\_FIFO/REG5/\$1I67: TBUF.HG.1;  
PLACE BLOCK PEND\_FIFO/REG5/\$1I69: TBUF.HG.2;  
PLACE BLOCK PEND\_FIFO/REG5/\$1I72: TBUF.IG.1;  
PLACE BLOCK PEND\_FIFO/REG4/R8BX7: EG;  
PLACE BLOCK PEND\_FIFO/REG4/R8BX5: FG;  
PLACE BLOCK PEND\_FIFO/REG4/R8BX3: GG;  
PLACE BLOCK PEND\_FIFO/REG4/R8BX1: HG;  
PLACE BLOCK PEND\_FIFO/REG4/\$1I24: TBUF.EH.2;  
PLACE BLOCK PEND\_FIFO/REG4/\$1I17: TBUF.FH.1;  
PLACE BLOCK PEND\_FIFO/REG4/\$1I10: TBUF.FH.2;  
PLACE BLOCK PEND\_FIFO/REG4/\$1I2: TBUF.GH.1;  
PLACE BLOCK PEND\_FIFO/REG4/\$1I65: TBUF.GH.2;  
PLACE BLOCK PEND\_FIFO/REG4/\$1I67: TBUF.HH.1;  
PLACE BLOCK PEND\_FIFO/REG4/\$1I69: TBUF.HH.2;  
PLACE BLOCK PEND\_FIFO/REG4/\$1I72: TBUF.IH.1;  
PLACE BLOCK PEND\_FIFO/REG3/R8BX7: EH;  
PLACE BLOCK PEND\_FIFO/REG3/R8BX5: FH;  
PLACE BLOCK PEND\_FIFO/REG3/R8BX3: GH;  
PLACE BLOCK PEND\_FIFO/REG3/R8BX1: HH;  
PLACE BLOCK PEND\_FIFO/REG3/\$1I24: TBUF.EI.2;  
PLACE BLOCK PEND\_FIFO/REG3/\$1I17: TBUF.FI.1;  
PLACE BLOCK PEND\_FIFO/REG3/\$1I10: TBUF.FI.2;  
PLACE BLOCK PEND\_FIFO/REG3/\$1I2: TBUF.GI.1;  
PLACE BLOCK PEND\_FIFO/REG3/\$1I65: TBUF.GL2;  
PLACE BLOCK PEND\_FIFO/REG3/\$1I67: TBUF.HI.1;  
PLACE BLOCK PEND\_FIFO/REG3/\$1I69: TBUF.HI.2;  
PLACE BLOCK PEND\_FIFO/REG3/\$1I72: TBUF.II.1;  
PLACE BLOCK PEND\_FIFO/REG2/R8BX7: EI;  
PLACE BLOCK PEND\_FIFO/REG2/R8BX5: FI;  
PLACE BLOCK PEND\_FIFO/REG2/R8BX3: GI;  
PLACE BLOCK PEND\_FIFO/REG2/R8BX1: HI;

PLACE BLOCK PEND\_FIFO/REG2/\$1I24: TBUF.EJ.2;  
PLACE BLOCK PEND\_FIFO/REG2/\$1I17: TBUF.FJ.1;  
PLACE BLOCK PEND\_FIFO/REG2/\$1I10: TBUF.FJ.2;  
PLACE BLOCK PEND\_FIFO/REG2/\$1I2: TBUF.GJ.1;  
PLACE BLOCK PEND\_FIFO/REG2/\$1I65: TBUF.GJ.2;  
PLACE BLOCK PEND\_FIFO/REG2/\$1I67: TBUF.HJ.1;  
PLACE BLOCK PEND\_FIFO/REG2/\$1I69: TBUF.HJ.2;  
PLACE BLOCK PEND\_FIFO/REG2/\$1I72: TBUF.IJ.1;  
PLACE BLOCK PEND\_FIFO/REG1/R8BX7: EJ;  
PLACE BLOCK PEND\_FIFO/REG1/R8BX5: FJ;  
PLACE BLOCK PEND\_FIFO/REG1/R8BX3: GJ;  
PLACE BLOCK PEND\_FIFO/REG1/R8BX1: HJ;  
PLACE BLOCK PEND\_FIFO/REG1/\$1I24: TBUF.EK.2;  
PLACE BLOCK PEND\_FIFO/REG1/\$1I17: TBUF.FK.1;  
PLACE BLOCK PEND\_FIFO/REG1/\$1I10: TBUF.FK.2;  
PLACE BLOCK PEND\_FIFO/REG1/\$1I2: TBUF.GK.1;  
PLACE BLOCK PEND\_FIFO/REG1/\$1I65: TBUF.GK.2;  
PLACE BLOCK PEND\_FIFO/REG1/\$1I67: TBUF.HK.1;  
PLACE BLOCK PEND\_FIFO/REG1/\$1I69: TBUF.HK.2;  
PLACE BLOCK PEND\_FIFO/REG1/\$1I72: TBUF.IK.1;  
PLACE BLOCK PEND\_FIFO/REG0/R8BX7: EK;  
PLACE BLOCK PEND\_FIFO/REG0/R8BX5: FK;  
PLACE BLOCK PEND\_FIFO/REG0/R8BX3: GK;  
PLACE BLOCK PEND\_FIFO/REG0/R8BX1: HK;  
PLACE BLOCK PEND\_FIFO/REG0/\$1I24: TBUF.EL.2;  
PLACE BLOCK PEND\_FIFO/REG0/\$1I17: TBUF.FL.1;  
PLACE BLOCK PEND\_FIFO/REG0/\$1I10: TBUF.FL.2;  
PLACE BLOCK PEND\_FIFO/REG0/\$1I2: TBUF.GL.1;  
PLACE BLOCK PEND\_FIFO/REG0/\$1I65: TBUF.GL.2;  
PLACE BLOCK PEND\_FIFO/REG0/\$1I67: TBUF.HL.1;  
PLACE BLOCK PEND\_FIFO/REG0/\$1I69: TBUF.HL.2;  
PLACE BLOCK PEND\_FIFO/REG0/\$1I72: TBUF.IL.1;  
PLACE BLOCK PEND\_FIFO/WRT\_PTR6: DE;  
PLACE BLOCK PEND\_FIFO/WRT\_PTR4: DG;  
PLACE BLOCK PEND\_FIFO/WRT\_PTR2: DI;  
PLACE BLOCK PEND\_FIFO/WRT\_PTR0: DK;  
PLACE BLOCK PEND\_FIFO/W2: DF;  
PLACE BLOCK PEND\_FIFO/W0: DH;  
PLACE BLOCK L1PTR FIFO/REG3/R2BX1: CF;  
PLACE BLOCK L1PTR FIFO/REG3/\$1I40: TBUF.CG.2;  
PLACE BLOCK L1PTR FIFO/REG3/\$1I41: TBUF.DG.1;  
PLACE BLOCK L1PTR FIFO/REG2/R2BX1: CG;  
PLACE BLOCK L1PTR FIFO/REG2/\$1I40: TBUF.CH.2;  
PLACE BLOCK L1PTR FIFO/REG2/\$1I41: TBUF.DH.1;  
PLACE BLOCK L1PTR FIFO/REG1/R2BX1: CH;  
PLACE BLOCK L1PTR FIFO/REG1/\$1I40: TBUF.CI.2;  
PLACE BLOCK L1PTR FIFO/REG1/\$1I41: TBUF.DI.1;  
PLACE BLOCK L1PTR FIFO/REG0/R2BX1: CI;  
PLACE BLOCK L1PTR FIFO/REG0/\$1I40: TBUF.CJ.2;  
PLACE BLOCK L1PTR FIFO/REG0/\$1I41: TBUF.DJ.1;  
PLACE BLOCK L1PTR FIFO/WRT\_PTR2: BF;  
PLACE BLOCK L1PTR FIFO/WRT\_PTR0: BI;  
PLACE BLOCK L1PTR FIFO/\$1N302: BG;  
PLACE BLOCK L1PTR FIFO/\$1N298: BH;  
PLACE BLOCK EMPTY FIFO/REG7/R4BX3: JD;

```

PLACE BLOCK EMPTYFIFO/REG7/R4BX1: KD;
PLACE BLOCK EMPTYFIFO/REG7/$1I24: TBUF.JE.2;
PLACE BLOCK EMPTYFIFO/REG7/$1I17: TBUF.KE.1;
PLACE BLOCK EMPTYFIFO/REG7/$1I10: TBUF.KE.2;
PLACE BLOCK EMPTYFIFO/REG7/$1I2: TBUF.LE.1;
PLACE BLOCK EMPTYFIFO/REG6/R4BX3: JE;
PLACE BLOCK EMPTYFIFO/REG6/R4BX1: KE;
PLACE BLOCK EMPTYFIFO/REG6/$1I24: TBUF.JF.2;
PLACE BLOCK EMPTYFIFO/REG6/$1I17: TBUF.KF.1;
PLACE BLOCK EMPTYFIFO/REG6/$1I10: TBUF.KF.2;
PLACE BLOCK EMPTYFIFO/REG6/$1I2: TBUF.LF.1;
PLACE BLOCK EMPTYFIFO/REG5/R4BX3: JF;
PLACE BLOCK EMPTYFIFO/REG5/R4BX1: KF;
PLACE BLOCK EMPTYFIFO/REG5/$1I24: TBUF.JG.2;
PLACE BLOCK EMPTYFIFO/REG5/$1I17: TBUF.KG.1;
PLACE BLOCK EMPTYFIFO/REG5/$1I10: TBUF.KG.2;
PLACE BLOCK EMPTYFIFO/REG5/$1I2: TBUF.LG.1;
PLACE BLOCK EMPTYFIFO/REG4/R4BX3: JG;
PLACE BLOCK EMPTYFIFO/REG4/R4BX1: KG;
PLACE BLOCK EMPTYFIFO/REG4/$1I24: TBUF.JH.2;
PLACE BLOCK EMPTYFIFO/REG4/$1I17: TBUF.KH.1;
PLACE BLOCK EMPTYFIFO/REG4/$1I10: TBUF.KH.2;
PLACE BLOCK EMPTYFIFO/REG4/$1I2: TBUF.LH.1;
PLACE BLOCK EMPTYFIFO/REG3/R4BX3: JH;
PLACE BLOCK EMPTYFIFO/REG3/R4BX1: KH;
PLACE BLOCK EMPTYFIFO/REG3/$1I24: TBUF.JI.2;
PLACE BLOCK EMPTYFIFO/REG3/$1I17: TBUF.KI.1;
PLACE BLOCK EMPTYFIFO/REG3/$1I10: TBUF.KI.2;
PLACE BLOCK EMPTYFIFO/REG3/$1I2: TBUF.LI.1;
PLACE BLOCK EMPTYFIFO/REG2/R4BX3: JI;
PLACE BLOCK EMPTYFIFO/REG2/R4BX1: KI;
PLACE BLOCK EMPTYFIFO/REG2/$1I24: TBUF.JJ.2;
PLACE BLOCK EMPTYFIFO/REG2/$1I17: TBUF.KJ.1;
PLACE BLOCK EMPTYFIFO/REG2/$1I10: TBUF.KJ.2;
PLACE BLOCK EMPTYFIFO/REG2/$1I2: TBUF.LJ.1;
PLACE BLOCK EMPTYFIFO/REG1/R4BX3: JJ;
PLACE BLOCK EMPTYFIFO/REG1/R4BX1: KJ;
PLACE BLOCK EMPTYFIFO/REG1/$1I24: TBUF.JK.2;
PLACE BLOCK EMPTYFIFO/REG1/$1I17: TBUF.KK.1;
PLACE BLOCK EMPTYFIFO/REG1/$1I10: TBUF.KK.2;
PLACE BLOCK EMPTYFIFO/REG1/$1I2: TBUF.LK.1;
PLACE BLOCK EMPTYFIFO/REG0/R4BX3: JK;
PLACE BLOCK EMPTYFIFO/REG0/R4BX1: KK;
PLACE BLOCK EMPTYFIFO/REG0/$1I24: TBUF.JL.2;
PLACE BLOCK EMPTYFIFO/REG0/$1I17: TBUF.KL.1;
PLACE BLOCK EMPTYFIFO/REG0/$1I10: TBUF.KL.2;
PLACE BLOCK EMPTYFIFO/REG0/$1I2: TBUF.LL.1;
PLACE BLOCK EMPTYFIFO/WRT_PTR6: IE;
PLACE BLOCK EMPTYFIFO/WRT_PTR4: IG;
PLACE BLOCK EMPTYFIFO/WRT_PTR2: II;
PLACE BLOCK EMPTYFIFO/WRT_PTR0: IK;
#
# pins from Ver1_2, Rev1
place block -ADDR1      : P36 ;
place block -ADDR2      : P40 ;

```

```

place block -E_FIFO_EMPTY      : P35 ;
place block -E_FIFO_FULL       : P25 ;
place block -E_FIFO_OE         : P46 ;
place block -EF_DATA           : P16 ;
place block -EMPTY_2_VME        : P47 ;
place block EVNTID0            : P19 ;
place block EVNTID1            : P20 ;
place block EVNTID2            : P18 ;
place block EVNTID3            : P17 ;
place block -L1_PFE             : P6 ;
place block -L1_PFF             : P8 ;
place block -L1_PTR0            : P70 ;
place block -L1_PTR1            : P71 ;
place block -L1A_ADD0            : P38 ;
place block -L1A_ADD1            : P45 ;
place block -L2A_ADD0            : P39 ;
place block -L2A_ADD1            : P44 ;
place block -P_FIFO_EMPTY       : P76 ;
place block -P_FIFO_FULL        : P79 ;
place block -P_FIFO_OE          : P62 ;
place block -PEND_2_VME          : P49 ;
place block -PTR_2_VME           : P52 ;
place block -PTR_DATA            : P27 ;
place block -RD_BM               : P29 ;
place block -RD_E_FIFO           : P28 ;
place block -RD_L1PF              : P9 ;
place block -RD_P_FIFO           : P80 ;
place block -SEL_BUF              : P69 ;
place block -VRBDAT_OE           : P61 ;
place block -WRT_BP               : P37 ;
place block -WRT_E_FIFO           : P26 ;
place block -WRT_L1PF              : P7 ;
place block -WRT_P_FIFO           : P81 ;
place block -WRT_P_REG             : P10 ;
place block BM_DAT0               : P30 ;
place block BM_DAT1               : P34 ;
place block BM_DAT2               : P24 ;
place block BM_DAT3               : P23 ;
place block READADD0              : P57 ;
place block READADD1              : P58 ;
place block READADD2              : P59 ;
place block READADD3              : P60 ;
place block VRB_DAT0              : P66 ;
place block VRB_DAT1              : P63 ;
place block VRB_DAT2              : P67 ;
place block VRB_DAT3              : P68 ;
place block -config2a             : P53 ;

```

# End

### **5.3 Placement Report**

PLACEMENT RESULTS FOR DESIGN BUF\_MNG  
From PPR Version 5.2.1

1997/06/03 03:17:02

Xilinx, Inc.  
(c) Copyright 1997. All Rights Reserved.

#### Report Contents

---

1. Device Utilization
2. Implementation Options

#### Device Utilization

---

##### Partitioned Design Utilization Using Part 3164APC84-5

---

	No. Used	Max Available	%Used
Occupied CLBs	147	224	65%
Bonded I/O Pins	50	70	71%
CLB Function Generators (*)	123	448	27%
CLB Flip Flops	157	448	35%
IOB Input Flip Flops	4	120	3%
IOB Output Flip Flops	16	120	13%
3-State Buffers	156	480	32%
3-State Longlines	26	32	81%

(\*) Each base F or FGM function counts as two

---

#### CPU Times

CPU time taken for Placement: 0 hrs 5 mins 17 secs

#### Implementation Options

---

### PPR Parameters

```
Design      = buf_mng
Parttype    = from design file
LogFile     = ppr.log
Outfile     = <design name>
Estimate    = FALSE
```

### Additional Specified or Non-Default Parameters

```
paramfile    = params.txt
cstfile     =
D:\CDF\SRC\Buf_mng\xproject\v1_4\rev1\Buf_mng.cst
seed         = 865307491
design       = buf_mng
placer_effort = 3
router_effort = 3
path_timing   = true
route_thru_bufg = ok
route_thru_blk = ok
guide_blk = all
lock_routing = whole_sigs
split_report = true
```

### Parameter Values from XACTINIT.DAT

```
ORCAD_NAMES      = false
```

```
===== End of Report =====
```

## **5.4 XACT Performance**

XACT PERFORMANCE RESULTS FOR DESIGN BUF\_MNG  
From PPR Version 5.2.1

1997/06/03 03:20:30

Xilinx, Inc.

(c) Copyright 1997. All Rights Reserved.

Xact Performance Summary

---

Parttype Used : 3164APC84

Speed Grade : -5

End-

Limit Actual Points

(ns) \* (ns) Missed Specification

---

<auto> 45.7 0/285 DEFAULT\_FROM\_FFS\_TO\_FFS=FROM:ffs:TO:ffs  
<auto> 55.4 0/402 DEFAULT\_FROM\_PADS\_TO\_FFS=FROM:pads:TO:ffs  
<auto> 56.9 0/20 DEFAULT\_FROM\_FFS\_TO\_PADS=FROM:ffs:TO:pads

(\*) Use the -FailedSpec and/or -TSMaxPaths options of the  
XDelay-TimeSpec command, accessible through the XDE or XDelay program,  
to confirm the actual path delays computed by PPR. Note that  
XDelay-TimeSpec will not report paths that start and end in the same  
block (CLB or IOB) and use no external routing.

===== End of Report =====



## 6. FIB Interface

### 6.1 Pin Description

Design				C D F / S R C / F I B _ I N T
Date				O c t o b e r 4 , 1 9 9 6
				3 / 2 0 / 9 8 2 : 1 6 P M
Signal	I	O	O T	I/O
Function				
<b>V M E A c c e s s</b>				
/F I B _ I N T	1			V M E Request line for F I B _ I N T
/B W R I T E	1			V M E Board write line
/S T R O B E [1:0]	2			V M E data strobes
/V M E O K 3		1		L o c a l a c k n o w l e d g e
V D A T 7 [9:0]			1 0	V M E data
V A D D R [4:1]	4			1 6 w o r d address space
<b>1 9</b>				
<b>H i s t o r y &amp; E m u l a t o r F i f o s</b>				
/R D _ F _ H [2:0]	3			R e a d F I B history fifo (individual)
/W R T _ F _ H	1			W r i t e F I B history fifos
/R E S _ F _ H	1			R e s e t F I B history fifos
/F _ H _ E F [2:0]	3			H i s t o r y f i f o E m p t y / F u l l f l a g
/F _ H _ H F [2:0]	3			H i s t o r y f i f o H a l f E m p t y / F u l l f l a g
/F _ H _ P A F E [2:0]	3			H i s t o r y f i f o A l m o s t F u l l / E m p t y f l a g
/F _ H 2 V M E [2:0]	3			H i s t o r y f i f o t o V M E
/R D _ F _ E	1			R e a d F I B e m u l a t o r f i f o
/W R T _ F _ E [1:0]	2			W r i t e F I B e m u l a t o r f i f o
/R E S _ F _ E	1			R e s e t F I B e m u l a t o r f i f o
/F _ E _ E F [1:0]	2			E m u l a t o r f i f o F u l l / E m p t y f l a g
/F _ E _ P A F E [1:0]	2			E m u l a t o r f i f o A l m o s t E m p t y / F u l l f l a g
/F _ E _ H F [1:0]	2			E m u l a t o r f i f o H a l f E m p t y / F u l l f l a g
<b>2 7</b>				
<b>G - L i n k C o n t r o l / S t a t u s</b>				
L V D S [3:1]	3			F i b G - L i n k s t a t u s & e r r o r l i n e s
L V D S 0	1			G - L i n k R U N / I N I T *
/G L _ D A V	1			G - L i n k D a t a V a l i d
/G L _ R E S	1			R e s e t F i n i s a r T x r e g i s t e r s
G L _ F F	1			T x f i l l f r a m e t y p e s e l e c t
G L _ E D	1			T x E n a b l e d a t a
G L _ L O C K	1			T x l o c k s t a t u s
/G L _ L O C K E D	1			G - L i n k c l o s e d l o o p l o c k i n d i c a t o r
<b>1 0</b>				
<b>C o n t r o l L i n e s f r o m T A X I</b>				
/E R R O R	1			S R C E r r o r t o T S I
/H A L T	1			C D F G l o b a l C o n t r o l
/R E C O V E R	1			C D F G l o b a l C o n t r o l
/R U N	1			C D F G l o b a l C o n t r o l
<b>4</b>				
<b>M i s c.</b>				
/X Q T	1			G l o b a l E x e c u t e l i n e
/T T L 2 E C L	1			R e s e t T T L / E C L t r a n s l a t o r s
/G L F E	1			G - L i n k f a t a l e r r o r t o E r r o r L o g g e r
/G L N F E	1			G - L i n k n o n - f a t a l e r r o r t o E r r o r L o g g e r
/S R C 2 F I B	1			S R C / F I B G - L i n k c o n n e c t i o n e s t a b l i s h e d
/E N _ R S M	1			E n a b l e R S M t o d r i v e F I B c o m m a n d l i n e s
R S M _ C L K	1			S t a t e M a c h i n e c l o c k
/C O N F I G 3		1		C o n f i g u r a t i o n i n d i c a t o r
<b>8</b>				
P i n c o u n t	3 3	2 5	0	1 0 6 8
6 8				

## 6.2 Constraint File

```
# Design: Fib_int
# Created by XACT Floorplanner ver 6.0.1
NOTPLACE BLOCK *: N2 C13 G14;
place block -BWRITE      : E12 ;
place block -EN_RSM       : J2 ;
place block -ERROR        : F1 ;
place block -F_E_EF0      : A12 ;
place block -F_E_EF1      : G13 ;
place block -F_E_HF0      : B12 ;
place block -F_E_HF1      : H14 ;
place block -F_E_PAFE0    : B11 ;
place block -F_E_PAFE1    : H13 ;
place block -F_H2VME0     : J13 ;
place block -F_H2VME1     : J12 ;
place block -F_H2VME2     : K12 ;
place block -F_H_EF0      : B5 ;
place block -F_H_EF1      : A4 ;
place block -F_H_EF2      : B7 ;
place block -F_H_HF1      : A3 ;
place block -F_H_PAFE1    : C5 ;
place block -F_H_PAFE2    : B6 ;
place block -FIB_INT      : D13 ;
place block -GL_DAV       : P10 ;
place block -GL_LOCKED    : P4 ;
place block -GL_RES       : P11 ;
place block -GLFE         : N5 ;
place block -GLNFE        : M6 ;
place block -HALT         : J3 ;
place block -RD_F_E       : G2 ;
place block -RD_F_H0      : K13 ;
place block -RD_F_H1      : L14 ;
place block -RD_F_H2      : L13 ;
place block -RECOVER       : K1 ;
place block -RES_F_E      : G1 ;
place block -RES_F_H      : F2 ;
place block -RUN           : H1 ;
place block -SRC2FIB      : N9 ;
place block -STROBE0      : C12 ;
place block -STROBE1      : F14 ;
place block -TTL2ECL      : J1 ;
place block -VMEOK3       : F13 ;
place block -WRT_F_H       : H2 ;
place block -WRT_FE0       : J14 ;
place block -WRT_FE1       : K14 ;
place block -XQT           : F3 ;
place block F_H_HF0        : A6 ;
place block F_H_PAFE0      : C6 ;
place block GL_ED          : M9 ;
place block GL_FF          : N10 ;
place block GL_LOCK        : P8 ;
place block LVDS0          : P9 ;
```

```
place block LVDS1      : N8 ;
place block LVDS2      : P6 ;
place block LVDS3      : P3 ;
place block N_CONFIG3   : C2 ;
place block N_F_H_HF2    : A5 ;
place block VADDR1      : E14 ;
place block VADDR2      : F12 ;
place block VADDR3      : E13 ;
place block VADDR4      : D14 ;
place block VDAT0       : C10 ;
place block VDAT1       : A11 ;
place block VDAT2       : B10 ;
place block VDAT3       : A10 ;
place block VDAT4       : C9 ;
place block VDAT5       : B9 ;
place block VDAT6       : A9 ;
place block VDAT7       : A8 ;
place block VDAT8       : B8 ;
place block VDAT9       : A7 ;
```

# End

### **6.3 Placement Report**

PLACEMENT RESULTS FOR DESIGN FIB\_INT  
From PPR Version 5.2.1

1997/06/30 21:57:06

Xilinx, Inc.  
(c) Copyright 1997. All Rights Reserved.

#### Report Contents

---

1. Device Utilization
2. Guided Placement Summary
3. Unguided Blocks from Input Design
4. Implementation Options

#### Device Utilization

---

##### Partitioned Design Utilization Using Part 3164APG132-3

---

	No.	Used	Max	Available	% Used
Occupied CLBs	163	224		72%	
Bonded I/O Pins	70	110		63%	
CLB Function Generators (*)	177	448		39%	
CLB Flip Flops	73	448		16%	
IOB Input Flip Flops	9	120		7%	
IOB Output Flip Flops	10	120		8%	
3-State Buffers	44	480		9%	
3-State Longlines	10	32		31%	

(\*) Each base F or FGM function counts as two

---

#### CPU Times

CPU time taken for Placement: 0 hrs 3 mins 48 secs

## Guided Placement Summary

---

The following table summarizes the guided placement process. For each type of LCA block, this table shows (1) how many blocks of that type exist in the input design file; (2) how many of these blocks were matched to blocks in the guide file; and (3) how many of the matched blocks were used to guide the placement of the input design.

A matched block will not be used to guide placement if any one of the following is true:

- \* The `guide_blk` option is set to `ROUTED_ONLY` and the matched block does not have any routing connected to it in the guide LCA file.
- \* The matched block is placed elsewhere by a constraint in the input design or the CST file.
- \* A different block is placed in the matched block's guide location by a constraint in the input design or the CST file.
- \* The matched block is part of an RLOC set and other blocks in that set were not matched. A partially-matched RLOC set will be guided only if there is space available for the unmatched blocks AND if the structure of the RLOC set can be respected.
- \* A pin on a matched block does not have any routing connected to it in the guide LCA file, so the pin was swapped with another pin on that CLB to improve the routing of the associated signal(s). For example, the X and Y flip-flops may be swapped if their output pins are not routed.

Block Type	In Design	Matched in Guide	Placement Guided
CLB flip-flop	73	72	51
CLB base FG function	128	61	59
CLB base F function	49	6	6
CLB base FGM function	0	0	0
3-state buffer	44	44	44
I/O pad	70	70	70

If there are blocks in the input design which could not be matched in the guide file they will be listed in the "Unguided Blocks" chapter below.

## Unguided Blocks from Input Design

---

The following lists show which blocks in the input design were not matched to blocks in the guide file and therefore were NOT guided during the placement process.

CLB flip flops driving following Q output signals:

---

WRT\_HF  
U202/SV3

U202/SV2  
U202/SV1  
U202/SV0  
RES\_HF  
RESTTLECL  
RD\_HF  
RD\_EF  
FIFOS\_OFF  
\$2N96  
U200/SV3  
U200/SV2  
U200/SV1  
U200/SV0  
GL\_STS4  
RES  
\$2N111  
FF  
ED  
DAV  
CLR\_GL\_CNT

CLB function generators driving following output signals:

-----  
Base Output Signal Name

-----  
F U202/WRT\_HF\$REG  
F U202/SV3\$REG  
F U202/SV1\$REG  
F U202/SV0\$REG  
F U202/RD\_HF\$REG  
F U202/RD\_EF\$REG  
F U202/DIS\_RSM\$REG  
F U200/SV2\$REG  
F U200/SV1\$REG  
F U200/SV0\$REG  
F U200/RES\$REG  
F U200/GLRUN\$REG  
F U202/\_93\_  
F U202/\_78\_  
F U202/\_77\_  
F U202/\_72\_  
F U202/\_69\_  
F U202/\_67\_  
F U202/\_58\_  
F U202/\_56\_  
F U202/\_175\_  
F U202/\_169\_  
F U202/\_165\_  
F U202/\_158\_  
F U202/\_156\_  
F U202/\_152\_  
F U202/\_150\_  
F U202/\_144\_  
F U202/\_143\_  
F U202/\_142\_

F U202/\_139\_  
F U202/\_115\_  
F U202/\_104\_  
F U200/\_89\_  
F U200/\_88\_  
F U200/\_74\_  
F U200/\_68\_  
F U200/\_62\_  
F U200/\_111\_  
F U200/\_110\_  
F U200/\_108\_  
F U200/\_106\_  
F U200/GLNFE\$REG  
FG U202/SV2\$REG  
FG U202/RES\_HF\$REG  
FG U202/RESTTLECL\$REG  
FG U202/FIFOS\_OFF\$REG  
FG U202/\_47\_  
FG U200/SV3\$REG  
FG U200/SRC2FIB\$REG  
FG U200/GLFE\$REG  
FG U200/FF\$REG  
FG U200/\_117\_  
FG U200/ED\$REG  
FG U200/CLR\_GL\_CNT\$REG  
FG U200/\_46\_  
FG U200/INC\_GL\_CNT\$REG  
FG U202/\_75\_  
FG U202/\_70\_  
FG U202/\_50\_  
FG U202/\_45\_  
FG U202/\_39\_  
FG U202/\_37\_  
FG U200/\_94\_  
FG U200/\_71\_  
FG U200/\_93\_  
FG U200/\_69\_  
FG U200/\_70\_  
FG U200/\_63\_  
FG U200/\_61\_  
FG U200/\_56\_  
FG U202/\_87\_  
FG U202/\_164\_  
FG U202/\_80\_  
FG U202/\_74\_  
FG U202/\_71\_  
FG U202/\_53\_  
FG U200/\_87\_  
FG U200/\_44\_  
FG U202/\_89\_  
FG U202/\_38\_  
FG U202/\_85\_  
FG U202/\_82\_  
FG U202/\_84\_  
FG U202/\_163\_

```
FG U202/_65_
FG U202/_145_
FG U202/_62_
FG U202/_48_
FG U202/_60_
FG U202/_57_
FG U202/_44_
FG U202/_155_
FG U200/_96_
FG U200/_41_
FG U200/_67_
FG U200/_39_
FG U200/_57_
FG U200/_40_
FG $1N1079
FG $1N1055
FG U202/_68_
FG U202/_59_
FG U202/_40_
FG U202/_181_
FG U202/_141_
FG U202/_140_
FG U202/RES_EF$REG
FG U200/_52_
FG U200/_48_
FG U200/_47_
FG U200/_101_
Implementation Options
```

---

#### PPR Parameters

```
Design      = fib_int
Parttype    = from design file
LogFile     = ppr.log
Outfile     = <design name>
Estimate    = FALSE
```

#### Additional Specified or Non-Default Parameters

```
paramfile    = params.txt
cstfile      =
D:\CDF\SRC\Fib_int\xproject\v1_0\rev1\Fib_int.cst
seed         = 867707580
design       = fib_int
placer_effort = 2
router_effort = 2
path_timing   = true
guide        =
d:\cdf\src\fib_int\xproject\v1_14\rev1\fib_int.lca
route_thru_bufg = ok
route_thru_blk = ok
guide_blk     = all
```

```
lock_routing      = whole_sigs  
split_report     = true
```

Parameter Values from XACTINIT.DAT

```
ORCAD_NAMES      = false
```

```
===== End of Report =====
```

## 6.4 XACT Performance

XACT PERFORMANCE RESULTS FOR DESIGN FIB\_INT  
From PPR Version 5.2.1

1997/06/30 22:07:19

Xilinx, Inc.

(c) Copyright 1997. All Rights Reserved.

Xact Performance Summary

---

Parttype Used : 3164APG132

Speed Grade : -3

End-

Limit	Actual	Points
(ns)	*	(ns) Missed Specification

---

<auto>	47.7	0/88	DEFAULT_FROM_FFS_TO_FFS=FROM:ffs:TO:ffs
<auto>	48.3	0/169	DEFAULT_FROM_PADS_TO_FFS=FROM:pads:TO:ffs
<auto>	45.1	0/34	DEFAULT_FROM_FFS_TO_PADS=FROM:ffs:TO:pads

(\*) Use the -FailedSpec and/or -TSMaxPaths options of the XDelay-TimeSpec command, accessible through the XDE or XDelay program, to confirm the actual path delays computed by PPR. Note that XDelay-TimeSpec will not report paths that start and end in the same block (CLB or IOB) and use no external routing.

Note: The design contains more than one clock signal (RF, 132NS, \$1N72, \$1N49). Default specs generated automatically by PPR will not control each clock at its own best possible frequency; clocks which could run faster may be controlled at a slower frequency. If you have specified no TIMESPEC requirements (or only a generic 'FROM:FFS:TO:FFS' requirement), the design performance may not have been optimized by PPR. For best results, supply individual TIMESPECS for the flip-flops on each clock signal that you care about.

\*\*\* PPR: WARNING 7028:

The design has flip-flops with asynchronous set/reset controls (PRE/SD or CLR/RD pins). When PPR analyzes design timing, it does not trace paths through the asynchronous set/reset input and on through the Q output.

If you want PPR to control the delay on paths through asynchronous set/reset pins, you must split the delay requirement into two segments: one ending at the set/reset input, and the other beginning at the flip-flop output. If you want PPR not to analyze paths that lead to asynchronous set/reset pins, attach an IGNORE specification to the pin(s) or signal(s).

By default, XDelay reports all paths through asynchronous set/reset pins. To prevent XDelay from showing these paths, use FlagBlk CLB\_Disable\_SR\_Q on the appropriate flip-flops.

===== End of Report =====

## 6.5 State CAD Diagrams

### 6.5.1 VME

/CDF/SRC/FIB\_INT/vme.dia  
Fib\_int FPGA VME interface  
19 Sept. 1996, JO  
UPDATED 14 MARCH, 1997

%READ% = !WRITE

%R0% = ADDR[0]

ADDR[] = 4:1

%R2% = ADDR[1]

REG0 = R0 & REQ

%R4% = ADDR[2]

REG2 = R2 & REQ

%R6% = ADDR[3]

REG4 = R4 & REQ

%R8% = ADDR[4]

REG6 = R6 & REQ

REG8 = R8 & FIFOs\_OFF & READ & REQ

%RA% = ADDR[5]

REGA = RA & FIFOs\_OFF & READ & REQ

%RC% = ADDR[6]

REGC = RC & WRITE & FIFOs\_OFF & REQ

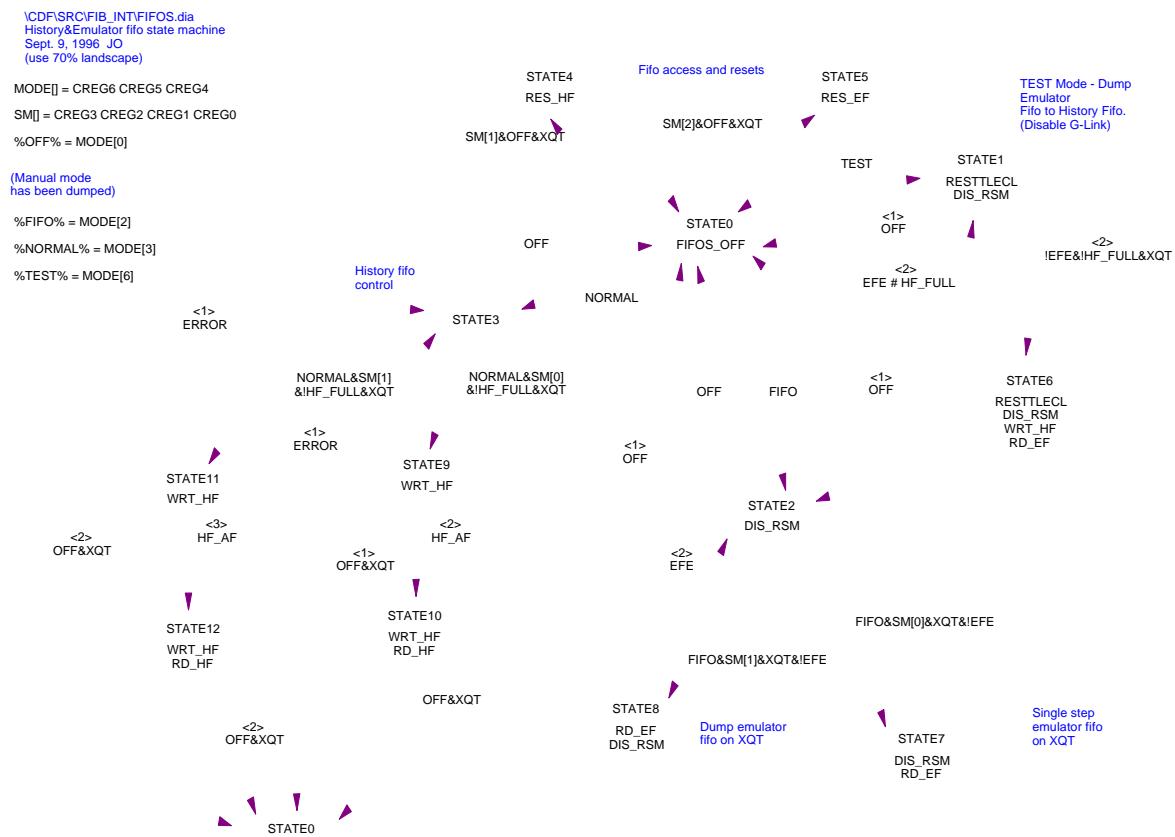
OK = REQ & (R0 or R4  
or (R2 & READ)  
or (R6 & READ)  
or (R8 & FIFOs\_OFF & READ)  
or (RA & FIFOs\_OFF & READ)  
or (RC & FIFOs\_OFF & WRITE))

PAD = OK & READ

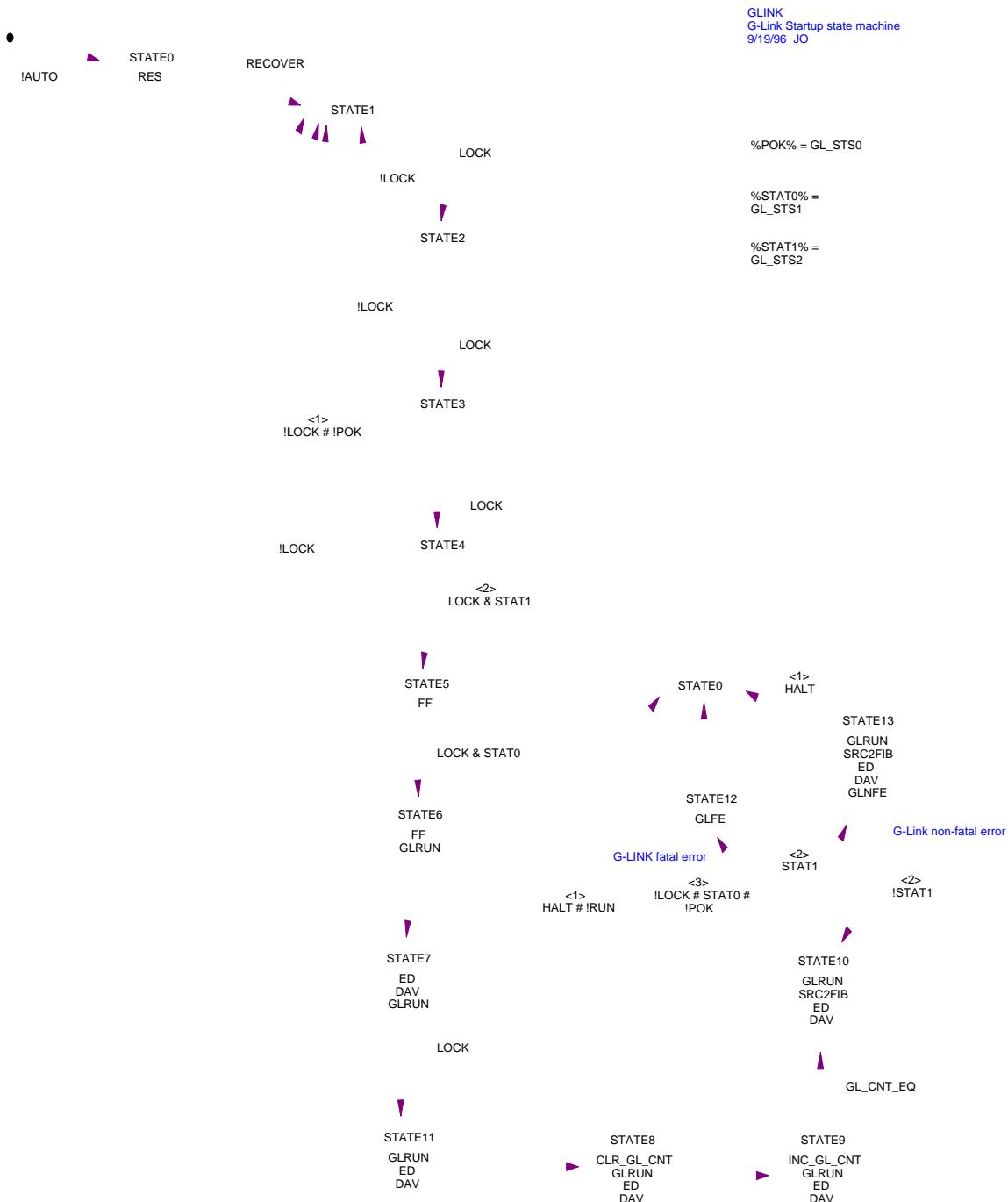
CHIP = OK & WRITE



## 6.5.2 FIFOs



### 6.5.3 GLINK



## 6.6 HDL Code

### 6.6.1 VME

```
" E:\HEPL\CDF\SRC\FIB_INT\VME.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.10
" Mon Jun 16 18:29:53 1997

" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are speed optimized.

MODULE VME

DECLARATIONS

"Input variables
    ADDR1 PIN;
    ADDR2 PIN;
    ADDR3 PIN;
    ADDR4 PIN;
    FIFOs_OFF PIN;
    REQ PIN;
    WRITE PIN;

"Logic variables
    CHIP PIN ISTYPE 'com';
    OK PIN ISTYPE 'com';
    PAD PIN ISTYPE 'com';
    REG0 PIN ISTYPE 'com';
    REG2 PIN ISTYPE 'com';
    REG4 PIN ISTYPE 'com';
    REG6 PIN ISTYPE 'com';
    REG8 PIN ISTYPE 'com';
    REGA PIN ISTYPE 'com';
    REGC PIN ISTYPE 'com';

"Vectors
DECLARATIONS
    ADDR=[  
        ADDR4,  
        ADDR3,  
        ADDR2,  
        ADDR1  
    ];

"Logic Equations
EQUATIONS
CHIP = OK & WRITE ;
```

```

OK = FIFOs_OFF & WRITE & !ADDR4 & ADDR2 & !ADDR1 & REQ # FIFOs_OFF & !
      WRITE & !ADDR4 & !ADDR2 & REQ # !WRITE & !ADDR4 & !ADDR3 & REQ # !ADDR4
&
      !ADDR3 & !ADDR1 & REQ ;

PAD = OK & !WRITE ;

REG0 = REQ & !ADDR4 & !ADDR3 & !ADDR2 & !ADDR1 ;

REG2 = REQ & !ADDR4 & !ADDR3 & !ADDR2 & ADDR1 ;

REG4 = REQ & !ADDR4 & !ADDR3 & ADDR2 & !ADDR1 ;

REG6 = REQ & !ADDR4 & !ADDR3 & ADDR2 & ADDR1 ;

REG8 = FIFOs_OFF & !WRITE & REQ & !ADDR4 & ADDR3 & !ADDR2 & !ADDR1 ;

REGA = FIFOs_OFF & !WRITE & REQ & !ADDR4 & ADDR3 & !ADDR2 & ADDR1 ;

REGC = WRITE & FIFOs_OFF & REQ & !ADDR4 & ADDR3 & ADDR2 & !ADDR1 ;

END VME

```

## 6.6.2 FIFOs

```
" D:\CDF\SRC\FIB_INT\FIFO.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.10
" Mon Jun 30 21:43:32 1997

" This Abel code was generated using:
" binary encoded state assignment with structured code format.
" Minimization is enabled, implied else is enabled,
" and outputs are speed optimized.
```

MODULE FIFOs

DECLARATIONS

```
"clock name
    CLK PIN;

"Input variables
    CREG0 PIN;
    CREG1 PIN;
    CREG2 PIN;
    CREG3 PIN;
    CREG4 PIN;
    CREG5 PIN;
    CREG6 PIN;
    EFE PIN;
    ERROR PIN;
    HF_AF PIN;
    HF_FULL PIN;
    XQT PIN;
```

"Output variables

```
    DIS_RSM PIN ISTYPE 'reg';
    FIFOs_OFF PIN ISTYPE 'reg';
    RD_EF PIN ISTYPE 'reg';
    RD_HF PIN ISTYPE 'reg';
    RES_EF PIN ISTYPE 'reg';
    RES_HF PIN ISTYPE 'reg';
    RESTTLECL PIN ISTYPE 'reg';
    WRT_HF PIN ISTYPE 'reg';
```

"State variables

```
    SV0 PIN ISTYPE 'reg';
    SV1 PIN ISTYPE 'reg';
    SV2 PIN ISTYPE 'reg';
    SV3 PIN ISTYPE 'reg';
```

"Vectors

DECLARATIONS

```
    MODE=[
        CREG6,
        CREG5,
```

```

        CREG4
    ];
    SM=[

        CREG3,
        CREG2,
        CREG1,
        CREG0
    ];

```

"Clocked logic clock setup

EQUATIONS

```

    DIS_RSM.clk=CLK;
    FIFOs_OFF.clk=CLK;
    RD_EF.clk=CLK;
    RD_HF.clk=CLK;
    RES_EF.clk=CLK;
    RES_HF.clk=CLK;
    RESTTLECL.clk=CLK;
    WRT_HF.clk=CLK;

```

"State Register assignment

DECLARATIONS

```

    sreg=[ SV0,SV1,SV2,SV3];

```

EQUATIONS

```

    sreg.clk=CLK;

```

DECLARATIONS

```

    STATE0=[0, 0, 0, 0];
    STATE1=[0, 0, 0, 1];
    STATE2=[0, 0, 1, 0];
    STATE3=[0, 0, 1, 1];
    STATE4=[0, 1, 0, 0];
    STATE5=[0, 1, 0, 1];
    STATE6=[0, 1, 1, 0];
    STATE7=[0, 1, 1, 1];
    STATE8=[1, 0, 0, 0];
    STATE9=[1, 0, 0, 1];
    STATE10=[1, 0, 1, 0];
    STATE11=[1, 0, 1, 1];
    STATE12=[1, 1, 0, 0];

```

state\_diagram sreg;

state STATE0:

```

    IF ( !CREG5 & !XQT # CREG4 & CREG6 # !CREG5 & CREG6 # !CREG5 & CREG4
        # !CREG5 & CREG3 # !CREG5 & CREG2 # !CREG5 & !CREG0 & !CREG1 #
    !CREG5
        & CREG1 & CREG0 ) THEN STATE0 WITH
        WRT_HF:=0;
        RESTTLECL:=0;
        RES_HF:=0;
        RES_EF:=0;

```

```

RD_HF:=0;
RD_EF:=0;
DIS_RSM:=0;
FIFOS_OFF:=1;
ENDWITH;
IF ( CREG6 & CREG5 & !CREG4 ) THEN STATE1 WITH
    WRT_HF:=0;
    RES_HF:=0;
    RES_EF:=0;
    RD_HF:=0;
    RD_EF:=0;
    FIFOS_OFF:=0;
    RESTTLECL:=1;
    DIS_RSM:=1;
ENDWITH;
IF ( !CREG6 & CREG5 & !CREG4 ) THEN STATE2 WITH
    WRT_HF:=0;
    RESTTLECL:=0;
    RES_HF:=0;
    RES_EF:=0;
    RD_HF:=0;
    RD_EF:=0;
    FIFOS_OFF:=0;
    DIS_RSM:=1;
ENDWITH;
IF ( !CREG6 & CREG5 & CREG4 ) THEN STATE3 WITH
    WRT_HF:=0;
    RESTTLECL:=0;
    RES_HF:=0;
    RES_EF:=0;
    RD_HF:=0;
    RD_EF:=0;
    FIFOS_OFF:=0;
    DIS_RSM:=0;
ENDWITH;
IF ( XQT & !CREG6 & !CREG5 & !CREG4 & !CREG3 & !CREG2 & !CREG1 & CREG0 )
    THEN STATE4 WITH
        WRT_HF:=0;
        RESTTLECL:=0;
        RES_EF:=0;
        RD_HF:=0;
        RD_EF:=0;
        FIFOS_OFF:=0;
        DIS_RSM:=0;
        RES_HF:=1;
ENDWITH;
IF ( XQT & !CREG6 & !CREG5 & !CREG4 & !CREG3 & !CREG2 & CREG1 & !CREG0 )
    THEN STATE5 WITH
        WRT_HF:=0;
        RESTTLECL:=0;
        RES_HF:=0;
        RD_HF:=0;
        RD_EF:=0;
        FIFOS_OFF:=0;
        DIS_RSM:=0;

```

```

        RES_EF:=1;
    ENDWITH;
state STATE1:
    IF ( !CREG6 & !CREG5 & !CREG4 ) THEN STATE0 WITH
        WRT_HF:=0;
        RESTTLECL:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        RD_EF:=0;
        DIS_RSM:=0;
        FIFOs_OFF:=1;
    ENDWITH;
    IF ( CREG6 & !EFE & !HF_FULL & XQT # CREG5 & !EFE & !HF_FULL & XQT #
        CREG4 & !EFE & !HF_FULL & XQT ) THEN STATE6 WITH
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        FIFOs_OFF:=0;
        RESTTLECL:=1;
        DIS_RSM:=1;
        WRT_HF:=1;
        RD_EF:=1;
    ENDWITH;
    IF ( EFE & CREG6 # HF_FULL & CREG6 # !XQT & CREG6 # EFE & CREG5 #
        HF_FULL & CREG5 # !XQT & CREG5 # EFE & CREG4 # HF_FULL & CREG4 #
        !XQT
        & CREG4 ) THEN STATE1 WITH
        WRT_HF:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        RD_EF:=0;
        FIFOs_OFF:=0;
        RESTTLECL:=1;
        DIS_RSM:=1;
    ENDWITH;
state STATE2:
    IF ( CREG5 & !XQT # CREG5 & EFE # CREG5 & CREG3 # CREG5 & CREG2 #
        CREG5 & CREG1 # CREG6 # CREG4 ) THEN STATE2 WITH
        WRT_HF:=0;
        RESTTLECL:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        RD_EF:=0;
        FIFOs_OFF:=0;
        DIS_RSM:=1;
    ENDWITH;
    IF ( !CREG6 & !CREG5 & !CREG4 ) THEN STATE0 WITH
        WRT_HF:=0;
        RESTTLECL:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;

```

```

        RD_EF:=0;
        DIS_RSM:=0;
        FIFOES_OFF:=1;
    ENDWITH;
    IF ( XQT & !EFE & !CREG3 & !CREG2 & !CREG1 & !CREG0 & !CREG6 & CREG5 & !
        CREG4 ) THEN STATE7 WITH
        WRT_HF:=0;
        RESTTLECL:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        FIFOES_OFF:=0;
        DIS_RSM:=1;
        RD_EF:=1;
    ENDWITH;
    IF ( XQT & !EFE & !CREG3 & !CREG2 & !CREG1 & CREG0 & !CREG6 & CREG5 &
!CREG4
        ) THEN STATE8 WITH
        WRT_HF:=0;
        RESTTLECL:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        FIFOES_OFF:=0;
        RD_EF:=1;
        DIS_RSM:=1;
    ENDWITH;
state STATE3:
    IF ( CREG4 & HF_FULL # CREG5 & HF_FULL # CREG4 & !XQT # CREG5 & !XQT
        # CREG4 & CREG3 # CREG5 & CREG3 # CREG4 & CREG2 # CREG5 & CREG2
    #
        CREG4 & CREG1 # CREG5 & CREG1 # CREG6 # CREG4 & !CREG5 # CREG5
    & !
        CREG4 ) THEN STATE3 WITH
        WRT_HF:=0;
        RESTTLECL:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        RD_EF:=0;
        FIFOES_OFF:=0;
        DIS_RSM:=0;
    ENDWITH;
    IF ( !CREG6 & !CREG5 & !CREG4 ) THEN STATE0 WITH
        WRT_HF:=0;
        RESTTLECL:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        RD_EF:=0;
        DIS_RSM:=0;
        FIFOES_OFF:=1;
    ENDWITH;
    IF ( !HF_FULL & XQT & !CREG3 & !CREG2 & !CREG1 & !CREG0 & !CREG6 & CREG5 &
        CREG4 ) THEN STATE9 WITH

```

```

        RESTTLECL:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        RD_EF:=0;
        FIFOS_OFF:=0;
        DIS_RSM:=0;
        WRT_HF:=1;
    ENDWITH;
    IF ( !HF_FULL & XQT & !CREG3 & !CREG2 & !CREG1 & CREG0 & !CREG6 & CREG5 &
        CREG4 ) THEN STATE11 WITH
        RESTTLECL:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        RD_EF:=0;
        FIFOS_OFF:=0;
        DIS_RSM:=0;
        WRT_HF:=1;
    ENDWITH;
state STATE4:
    GOTO STATE0 WITH
        WRT_HF:=0;
        RESTTLECL:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        RD_EF:=0;
        DIS_RSM:=0;
        FIFOS_OFF:=1;
    ENDWITH;
state STATE5:
    GOTO STATE0 WITH
        WRT_HF:=0;
        RESTTLECL:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        RD_EF:=0;
        DIS_RSM:=0;
        FIFOS_OFF:=1;
    ENDWITH;
state STATE6:
    IF ( !CREG6 & !CREG5 & !CREG4 ) THEN STATE0 WITH
        WRT_HF:=0;
        RESTTLECL:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        RD_EF:=0;
        DIS_RSM:=0;
        FIFOS_OFF:=1;
    ENDWITH;
    IF ( CREG6 & HF_FULL # CREG5 & HF_FULL # CREG4 & HF_FULL # CREG6 & EFE
        # CREG5 & EFE # CREG4 & EFE ) THEN STATE1 WITH

```

```

        WRT_HF:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        RD_EF:=0;
        FIFOS_OFF:=0;
        RESTTLECL:=1;
        DIS_RSM:=1;
    ENDWITH;
    IF ( !HF_FULL & !EFE & CREG6 # !HF_FULL & !EFE & CREG5 # !HF_FULL & !EFE
        & CREG4 ) THEN STATE6 WITH
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        FIFOS_OFF:=0;
        RESTTLECL:=1;
        DIS_RSM:=1;
        WRT_HF:=1;
        RD_EF:=1;
    ENDWITH;
state STATE7:
    GOTO STATE2 WITH
        WRT_HF:=0;
        RESTTLECL:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        RD_EF:=0;
        FIFOS_OFF:=0;
        DIS_RSM:=1;
    ENDWITH;
state STATE8:
    IF ( !CREG6 & !CREG5 & !CREG4 ) THEN STATE0 WITH
        WRT_HF:=0;
        RESTTLECL:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        RD_EF:=0;
        DIS_RSM:=0;
        FIFOS_OFF:=1;
    ENDWITH;
    IF ( CREG6 & EFE # CREG5 & EFE # CREG4 & EFE ) THEN STATE2 WITH
        WRT_HF:=0;
        RESTTLECL:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        RD_EF:=0;
        FIFOS_OFF:=0;
        DIS_RSM:=1;
    ENDWITH;
    IF ( !EFE & CREG6 # !EFE & CREG5 # !EFE & CREG4 ) THEN STATE8 WITH
        WRT_HF:=0;
        RESTTLECL:=0;

```

```

    RES_HF:=0;
    RES_EF:=0;
    RD_HF:=0;
    FIFOs_OFF:=0;
    RD_EF:=1;
    DIS_RSM:=1;
  ENDWITH;
state STATE9:
  IF ( XQT & !CREG6 & !CREG5 & !CREG4 ) THEN STATE0 WITH
    WRT_HF:=0;
    RESTTLECL:=0;
    RES_HF:=0;
    RES_EF:=0;
    RD_HF:=0;
    RD_EF:=0;
    DIS_RSM:=0;
    FIFOs_OFF:=1;
  ENDWITH;
  IF ( !XQT & HF_AF # CREG6 & HF_AF # CREG5 & HF_AF # CREG4 & HF_AF )
    THEN STATE10 WITH
      RESTTLECL:=0;
      RES_HF:=0;
      RES_EF:=0;
      RD_EF:=0;
      FIFOs_OFF:=0;
      DIS_RSM:=0;
      WRT_HF:=1;
      RD_HF:=1;
    ENDWITH;
  IF ( !HF_AF & !XQT # !HF_AF & CREG6 # !HF_AF & CREG5 # !HF_AF & CREG4
    ) THEN STATE9 WITH
    RESTTLECL:=0;
    RES_HF:=0;
    RES_EF:=0;
    RD_HF:=0;
    RD_EF:=0;
    FIFOs_OFF:=0;
    DIS_RSM:=0;
    WRT_HF:=1;
  ENDWITH;
state STATE10:
  IF ( XQT & !CREG6 & !CREG5 & !CREG4 ) THEN STATE0 WITH
    WRT_HF:=0;
    RESTTLECL:=0;
    RES_HF:=0;
    RES_EF:=0;
    RD_HF:=0;
    RD_EF:=0;
    DIS_RSM:=0;
    FIFOs_OFF:=1;
  ENDWITH;
  IF ( CREG4 # CREG5 # CREG6 # !XQT ) THEN STATE10 WITH
    RESTTLECL:=0;
    RES_HF:=0;
    RES_EF:=0;

```

```

RD_EF:=0;
FIFOS_OFF:=0;
DIS_RSM:=0;
WRT_HF:=1;
RD_HF:=1;
ENDWITH;
state STATE11:
IF ( ERROR ) THEN STATE3 WITH
    WRT_HF:=0;
    RESTTLECL:=0;
    RES_HF:=0;
    RES_EF:=0;
    RD_HF:=0;
    RD_EF:=0;
    FIFOS_OFF:=0;
    DIS_RSM:=0;
ENDWITH;
IF ( !ERROR & XQT & !CREG6 & !CREG5 & !CREG4 ) THEN STATE0 WITH
    WRT_HF:=0;
    RESTTLECL:=0;
    RES_HF:=0;
    RES_EF:=0;
    RD_HF:=0;
    RD_EF:=0;
    DIS_RSM:=0;
    FIFOS_OFF:=1;
ENDWITH;
IF ( !ERROR & CREG4 & HF_AF # !ERROR & CREG5 & HF_AF # !ERROR & CREG6 &
    HF_AF # !ERROR & !XQT & HF_AF ) THEN STATE12 WITH
    RESTTLECL:=0;
    RES_HF:=0;
    RES_EF:=0;
    RD_EF:=0;
    FIFOS_OFF:=0;
    DIS_RSM:=0;
    WRT_HF:=1;
    RD_HF:=1;
ENDWITH;
IF ( !HF_AF & !XQT & !ERROR # !HF_AF & CREG6 & !ERROR # !HF_AF & CREG5 &
    !ERROR # !HF_AF & CREG4 & !ERROR ) THEN STATE11 WITH
    RESTTLECL:=0;
    RES_HF:=0;
    RES_EF:=0;
    RD_HF:=0;
    RD_EF:=0;
    FIFOS_OFF:=0;
    DIS_RSM:=0;
    WRT_HF:=1;
ENDWITH;
state STATE12:
IF ( ERROR ) THEN STATE3 WITH
    WRT_HF:=0;
    RESTTLECL:=0;
    RES_HF:=0;
    RES_EF:=0;

```

```

        RD_HF:=0;
        RD_EF:=0;
        FIFOES_OFF:=0;
        DIS_RSM:=0;
    ENDWITH;
    IF ( !ERROR & XQT & !CREG6 & !CREG5 & !CREG4 ) THEN STATE0 WITH
        WRT_HF:=0;
        RESTTLECL:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_HF:=0;
        RD_EF:=0;
        DIS_RSM:=0;
        FIFOES_OFF:=1;
    ENDWITH;
    IF ( CREG4 & !ERROR # CREG5 & !ERROR # CREG6 & !ERROR # !XQT & !ERROR
        ) THEN STATE12 WITH
        RESTTLECL:=0;
        RES_HF:=0;
        RES_EF:=0;
        RD_EF:=0;
        FIFOES_OFF:=0;
        DIS_RSM:=0;
        WRT_HF:=1;
        RD_HF:=1;
    ENDWITH;
END FIFOS

```

### 6.6.3 GLINK

```
" E:\HEPL\CDF\SRC\FIB_INT\GLINK.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.10
" Fri Jun 13 12:05:55 1997

" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are speed optimized.
```

MODULE GLINK

DECLARATIONS

```
"clock name
    CLK PIN;
```

"Input variables

```
    AUTO PIN;
    GL_CNT_EQ PIN;
    GL_STS0 PIN;
    GL_STS1 PIN;
    GL_STS2 PIN;
    HALT PIN;
    LOCK PIN;
    RECOVER PIN;
    RUN PIN;
```

"Output variables

```
    CLR_GL_CNT PIN ISTYPE 'reg';
    DAV PIN ISTYPE 'reg';
    ED PIN ISTYPE 'reg';
    FF PIN ISTYPE 'reg';
    GLFE PIN ISTYPE 'reg';
    GLNFE PIN ISTYPE 'reg';
    GLRUN PIN ISTYPE 'reg';
    INC_GL_CNT PIN ISTYPE 'reg';
    RES PIN ISTYPE 'reg';
    SRC2FIB PIN ISTYPE 'reg';
```

"State variables

```
    SV0 PIN ISTYPE 'reg';
    SV1 PIN ISTYPE 'reg';
    SV2 PIN ISTYPE 'reg';
    SV3 PIN ISTYPE 'reg';
```

"Clocked logic clock setup

EQUATIONS

```
    CLR_GL_CNT.clk=CLK;
    DAV.clk=CLK;
    ED.clk=CLK;
```

```

FF.clk=CLK;
GLFE.clk=CLK;
GLNFE.clk=CLK;
GLRUN.clk=CLK;
INC_GL_CNT.clk=CLK;
RES.clk=CLK;
SRC2FIB.clk=CLK;

"State Register assignment
DECLARATIONS
    sreg=[ SV0,SV1,SV2,SV3];

EQUATIONS
    sreg.clk=CLK;

DECLARATIONS
    STATE0=[0, 0, 0, 0];
    STATE1=[0, 0, 0, 1];
    STATE2=[0, 0, 1, 0];
    STATE3=[0, 0, 1, 1];
    STATE4=[0, 1, 0, 0];
    STATE5=[0, 1, 0, 1];
    STATE6=[0, 1, 1, 0];
    STATE7=[0, 1, 1, 1];
    STATE8=[1, 0, 0, 0];
    STATE9=[1, 0, 0, 1];
    STATE10=[1, 0, 1, 0];
    STATE11=[1, 0, 1, 1];
    STATE12=[1, 1, 0, 0];
    STATE13=[1, 1, 0, 1];

EQUATIONS
    SV0 := AUTO & SV0.FB & !SV1.FB & !SV2.FB # AUTO & SV0.FB & !SV1.FB & SV3.FB
        # AUTO & LOCK & !SV0.FB & SV1.FB & SV2.FB & SV3.FB # AUTO & !HALT & RUN
        & SV0.FB & !SV1.FB # !HALT & AUTO & SV0.FB & !SV2.FB & SV3.FB ;
    SV1 := AUTO & LOCK & !SV0.FB & !SV1.FB & SV2.FB & SV3.FB # AUTO & LOCK &
        GL_STS0 & !SV0.FB & SV1.FB & !SV3.FB # AUTO & LOCK & !SV0.FB & SV1.FB & !
        SV2.FB & SV3.FB # AUTO & !SV0.FB & SV1.FB & SV2.FB & !SV3.FB # !LOCK &
        AUTO & !SV0.FB & SV1.FB & SV2.FB # AUTO & !HALT & RUN & !GL_STS0 & SV0.FB
        &
        !SV1.FB & SV2.FB & !SV3.FB # AUTO & !HALT & RUN & GL_STS1 & SV0.FB & !
        SV1.FB & SV2.FB & !SV3.FB # AUTO & !HALT & RUN & !LOCK & SV0.FB & !SV1.FB &
        SV2.FB & !SV3.FB # AUTO & !HALT & RUN & GL_STS2 & SV0.FB & !SV1.FB &
        SV2.FB & !SV3.FB # GL_STS2 & !HALT & AUTO & SV0.FB & SV1.FB & !SV2.FB &
        SV3.FB ;
    SV2 := AUTO & LOCK & !SV0.FB & !SV1.FB & !SV2.FB & SV3.FB # AUTO & LOCK & !
        SV0.FB & SV2.FB & !SV3.FB # AUTO & LOCK & GL_STS1 & !SV0.FB & !SV2.FB &
        SV3.FB # AUTO & GL_CNT_EQ & SV0.FB & !SV1.FB & !SV2.FB & SV3.FB # GL_STS0
        & !GL_STS1 & LOCK & !GL_STS2 & RUN & !HALT & AUTO & !SV1.FB & SV2.FB & !
        SV3.FB # AUTO & !HALT & !GL_STS2 & SV0.FB & SV1.FB & !SV2.FB & SV3.FB #
        AUTO & !SV0.FB & SV1.FB & SV2.FB ;

```

```

SV3 := AUTO & RECOVER & !SV1.FB & !SV2.FB & !SV3.FB # AUTO & !LOCK &
    SV0.FB & !SV1.FB & SV2.FB # AUTO & !GL_STS0 & !SV0.FB & SV1.FB & !SV3.FB #
        AUTO & !LOCK & !SV0.FB & SV1.FB & !SV2.FB # AUTO & GL_STS2 & !SV0.FB &
        SV1.FB & !SV3.FB # !GL_STS1 & AUTO & !SV0.FB & SV1.FB & !SV2.FB & SV3.FB #
            AUTO & !SV0.FB & SV2.FB & !SV3.FB # !LOCK & AUTO & !SV0.FB & SV3.FB #
                AUTO & SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB # !GL_CNT_EQ & AUTO & SV0.FB &
                !SV1.FB & !SV2.FB # AUTO & LOCK & !SV0.FB & SV1.FB & SV2.FB # AUTO & !
                HALT & RUN & GL_STS2 & !SV1.FB & SV2.FB & !SV3.FB # GL_STS2 & !HALT & AUTO
                & SV0.FB & SV1.FB & !SV2.FB & SV3.FB ;

CLR_GL_CNT := AUTO & SV0.FB & !SV1.FB & SV2.FB & SV3.FB ;

DAV := AUTO & SV0.FB & !SV1.FB & !SV2.FB # AUTO & SV0.FB & !SV1.FB & SV3.FB
    # GL_STS0 & !GL_STS1 & LOCK & RUN & !HALT & AUTO & SV0.FB & !SV1.FB #
        AUTO & !SV0.FB & SV1.FB & SV2.FB # AUTO & !HALT & RUN & GL_STS2 & SV0.FB &
        !SV1.FB # !HALT & AUTO & SV0.FB & !SV2.FB & SV3.FB ;

ED := AUTO & SV0.FB & !SV1.FB & !SV2.FB # AUTO & SV0.FB & !SV1.FB & SV3.FB
    # GL_STS0 & !GL_STS1 & LOCK & RUN & !HALT & AUTO & SV0.FB & !SV1.FB #
        AUTO & !SV0.FB & SV1.FB & SV2.FB # AUTO & !HALT & RUN & GL_STS2 & SV0.FB &
        !SV1.FB # !HALT & AUTO & SV0.FB & !SV2.FB & SV3.FB ;

FF := AUTO & LOCK & GL_STS0 & GL_STS2 & !SV0.FB & SV1.FB & !SV2.FB # AUTO &
    LOCK & !SV0.FB & SV1.FB & !SV2.FB & SV3.FB ;

GLFE := AUTO & !HALT & RUN & !GL_STS2 & !GL_STS0 & SV0.FB & !SV1.FB & SV2.FB
    & !SV3.FB # AUTO & !HALT & RUN & !GL_STS2 & GL_STS1 & SV0.FB & !SV1.FB &
    SV2.FB & !SV3.FB # AUTO & !HALT & RUN & !GL_STS2 & !LOCK & SV0.FB & !SV1.FB
    & SV2.FB & !SV3.FB ;

GLNFE := AUTO & !HALT & RUN & GL_STS2 & SV0.FB & !SV1.FB & SV2.FB & !SV3.FB
    # GL_STS2 & !HALT & AUTO & SV0.FB & SV1.FB & !SV2.FB & SV3.FB ;

GLRUN := AUTO & LOCK & GL_STS1 & !SV0.FB & SV1.FB & SV3.FB # AUTO & SV0.FB
    & !SV1.FB & !SV2.FB # AUTO & SV0.FB & !SV1.FB & SV3.FB # GL_STS0 & !
        GL_STS1 & LOCK & RUN & !HALT & AUTO & SV0.FB & !SV1.FB # AUTO & !SV0.FB &
        SV1.FB & SV2.FB # AUTO & !HALT & RUN & GL_STS2 & SV0.FB & !SV1.FB # !HALT
        & AUTO & SV0.FB & !SV2.FB & SV3.FB ;

INC_GL_CNT := AUTO & SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB # !GL_CNT_EQ &
    AUTO & SV0.FB & !SV1.FB & !SV2.FB ;

RES := !RECOVER & !SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB # !RUN & SV0.FB & !
    SV1.FB & SV2.FB & !SV3.FB # HALT & SV0.FB & !SV1.FB & SV2.FB & !SV3.FB #
    SV0.FB & SV1.FB & !SV2.FB & !SV3.FB # HALT & SV0.FB & SV1.FB & !SV2.FB #
    !AUTO ;

SRC2FIB := AUTO & GL_CNT_EQ & SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB # GL_STS0
    & !GL_STS1 & LOCK & RUN & !HALT & AUTO & SV0.FB & !SV1.FB & SV2.FB &
    !SV3.FB
        # AUTO & !HALT & RUN & GL_STS2 & SV0.FB & !SV1.FB & SV2.FB & !SV3.FB # !
            HALT & AUTO & SV0.FB & SV1.FB & !SV2.FB & SV3.FB ;
END GLINK

```

## 7. Taxi Decoder

### 7.1 Pin Description

Design					Taxi_dec
Date					Nov. 11, 1996
					7/21/97 11:42
Signal	I	O	OT	I/O	Function
					<b>Taxi chip outputs</b>
TXCSTRB	1				Taxi command strobe
TXDSTRB	1				Taxi data strobe
TXVLTN	1				Taxi chip violation
TX[8:0]]	9				Taxi output data
<b>12</b>					
					<b>Decoded TSI Commands</b>
/TEST				1	Test sequence
/RUN				1	DAQ RUN mode
/RECOVER				1	DAQ RECOVER mode
/HALT				1	DAQ HALT mode
/CONTROL				1	Control word
/CALIB				1	Calibration word
CALIB[2:0]				3	Calibration word data
EID[3:0]				4	Event ID
/L1A				1	Level 1 Accept
L1A[1:0]				2	L1 Accept pointer
/L2A				1	L2 Accept
L2A[1:0]				2	L2 Accept pointer
/L2R				1	L2 Reject
L2R[1:0]				2	L2 Reject pointer
<b>22</b>					
/U_DOIT	1				Allows TAXI_DEC control of output bus
RSM_CLK	1				Board level state machine clock (gclk)
RSM_CLK_DEL	1				Delayed state machine clock (aclk)
	3				
	15	0	0	22	<b>37</b>
<b>37</b>					

## 7.2 Constraint File

```
# Design: Taxi_dec
# Created by XACT Floorplanner ver 6.0.1
PLACE BLOCK TXDSTRB: P26;
PLACE BLOCK TXCSTRB: P27;
PLACE BLOCK -U_DOIT: P75;
PLACE BLOCK -L1A: P10;
PLACE BLOCK TX8: P24;
PLACE BLOCK TX7: P23;
PLACE BLOCK TX6: P21;
PLACE BLOCK TX5: P20;
PLACE BLOCK TX4: P19;
PLACE BLOCK TX3: P18;
PLACE BLOCK TX2: P17;
PLACE BLOCK TX1: P16;
PLACE BLOCK TX0: P15;
PLACE BLOCK L2R_ADD1: AC;
PLACE BLOCK L2A_ADD1: CC;
PLACE BLOCK L1_ADD1: AB;
PLACE BLOCK $1I40/FR20: BA;
PLACE BLOCK $1I40/FR10: CA;
PLACE BLOCK $1N197: AA;
PLACE BLOCK $1N179: BB;
PLACE BLOCK $1N161: BC;
PLACE BLOCK $1N119: CD;
PLACE BLOCK $1N95: DD;
PLACE BLOCK EVNTID2: EC;
PLACE BLOCK EVNTID0: DC;
PLACE BLOCK CAL2: CE;
PLACE BLOCK CAL0: BD;
PLACE BLOCK $1I40/FR25: EA;
PLACE BLOCK $1I40/FR22: DA;
PLACE BLOCK $1I40/FR18: FA;
PLACE BLOCK $1I40/FR16: EB;
PLACE BLOCK $1I40/FR15: DB;
PLACE BLOCK $1I40/FR12: CB;
PLACE BLOCK $1N83: DE;
PLACE BLOCK $1N77: ED;
PLACE BLOCK L2R1: P5;
PLACE BLOCK L2R0: P6;
PLACE BLOCK L2A1: P2;
PLACE BLOCK L2A0: P3;
PLACE BLOCK L1A1: P8;
PLACE BLOCK L1A0: P9;
PLACE BLOCK EID3: P81;
PLACE BLOCK EID2: P82;
PLACE BLOCK EID1: P83;
PLACE BLOCK EID0: P84;
PLACE BLOCK CALIB2: P77;
PLACE BLOCK CALIB1: P78;
PLACE BLOCK CALIB0: P79;
PLACE BLOCK -TEST: P69;
```

PLACE BLOCK -RUN: P70;  
PLACE BLOCK -RECOVER: P71;  
PLACE BLOCK -HALT: P68;  
PLACE BLOCK -CONT: P73;  
PLACE BLOCK -CAL: P80;  
PLACE BLOCK -L2R: P7;  
PLACE BLOCK -L2A: P4;  
# End

### **7.3 Placement Report**

PLACEMENT RESULTS FOR DESIGN TAXI\_DEC  
From PPR Version 5.2.1

Ver1.1 Rev2

1996/11/14 14:50:48

Xilinx, Inc.

(c) Copyright 1996. All Rights Reserved.

#### Report Contents

---

1. Device Utilization
2. Implementation Options

#### Device Utilization

---

##### Partitioned Design Utilization Using Part 3130APC84-2

---

	No. Used	Max Available	%Used
Occupied CLBs	27	100	27%
Bonded I/O Pins	36	74	48%
CLB Function Generators (*)	7	200	3%
CLB Flip Flops	40	200	20%
IOB Input Flip Flops	0	80	0%
IOB Output Flip Flops	22	80	27%
3-State Buffers	0	220	0%
3-State Longlines	0	20	0%

(\*) Each base F or FGM function counts as two

---

#### CPU Times

CPU time taken for Placement: 0 hrs 0 mins 13 secs

#### Implementation Options

---

### PPR Parameters

```
Design      = taxi_dec
Parttype    = from design file
LogFile     = ppr.log
Outfile     = <design name>
Estimate    = FALSE
```

### Additional Specified or Non-Default Parameters

```
paramfile    = params.txt
cstfile      =
D:\CDF\SRC\taxi_dec\xproject\v1_1\rev1\Taxi_dec.cst
seed         = 847983025
design       = taxi_dec
placer_effort = 3
router_effort = 3
path_timing   = true
route_thru_bufg = ok
route_thru_blk = ok
guide_blk    = all
lock_routing  = whole_sigs
split_report  = true
```

### Parameter Values from XACTINIT.DAT

```
ORCAD_NAMES      = false
```

```
===== End of Report =====
```

## 7.4 XACT Performance

XACT PERFORMANCE RESULTS FOR DESIGN TAXI\_DEC

From PPR Version 5.2.1

1996/11/14 14:51:10

Xilinx, Inc.

(c) Copyright 1996. All Rights Reserved.

Xact Performance Summary

---

Parttype Used : 3130APC84

Speed Grade : -2

End-

Limit	Actual	Points
(ns)	*	(ns) Missed Specification

---

<auto>	18.4	0/61	DEFAULT_FROM_FFS_TO_FFS=FROM:ffs:TO:ffs
<auto>	5.4	0/80	DEFAULT_FROM_PADS_TO_FFS=FROM:pads:TO:ffs
<auto>	3.7	0/22	DEFAULT_FROM_FFS_TO_PADS=FROM:ffs:TO:pads

(\*) Use the -FailedSpec and/or -TSMaxPaths options of the XDelay-TimeSpec command, accessible through the XDE or XDelay program, to confirm the actual path delays computed by PPR. Note that XDelay-TimeSpec will not report paths that start and end in the same block (CLB or IOB) and use no external routing.

Note: The design contains flip-flops clocked by opposite edges of the same signal. Default specs generated automatically by PPR will not properly control paths between opposite-edge flip-flops. The delays on such paths will be properly controlled ONLY if: (1) the flip-flops are grouped by clock edge using RISING and FALLING TIMEGRP statements, and these subgroups are used in TIMESPEC statements with the appropriate High and Low clock times; or (2) a C2S-style TIMESPEC statement with the appropriate High clock time is attached to the clock signal.

Note: The design contains more than one clock signal (\$1N56, \$1N58). Default specs generated automatically by PPR will not control each clock at its own best possible frequency; clocks which could run faster may be controlled at a slower frequency. If you have specified no TIMESPEC requirements (or only a generic 'FROM:FFS:TO:FFS' requirement), the design performance may not have been optimized by PPR. For best results, supply individual TIMESPECs for the flip-flops on each clock signal that you care about.

\*\*\* PPR: WARNING 7028:

The design has flip-flops with asynchronous set/reset controls (PRE/SD or CLR/RD pins). When PPR analyzes design timing, it does not trace paths through the asynchronous set/reset input and on through the Q output.

If you want PPR to control the delay on paths through asynchronous set/reset pins, you must split the delay requirement into two segments: one ending at the set/reset input, and the other beginning at the flip-flop output. If you want PPR not to analyze paths that lead to asynchronous set/reset pins, attach an IGNORE specification to the pin(s) or signal(s).

By default, XDelay reports all paths through asynchronous set/reset pins. To prevent XDelay from showing these paths, use FlagBlk CLB\_Disable\_SR\_Q on the appropriate flip-flops.

===== End of Report =====

## 8. TSI Emulator

### 8.1 Pin Description

Design					CDF/SRC/TAXI_EM
Date	I	O	O T	I/ O	Function
					January 23, 1997
					7/21/97 11:38 AM
Signal	I	O	O T	I/ O	Function
<b>VME Access</b>					<b>VME Access</b>
/TAXI	1				VME Request line for taxi
/BWRITE	1				VME Board write line
/STROBE[1:0]	2				VME data strobes
/VMEOK4		1			Local acknowledge
VDAT[15:0]				16	VME data
VADDR[4:1]	4				16 word address space
<b>25</b>					
Emulator FIFO Controls					Emulator FIFO Controls
/RES_TX_E		1			Taxi Emulator FIFO Reset
/WRT_TX_E		1			Write to Taxi Emulator FIFO
/RD_TX_E		1			Read from Taxi Emulator FIFO
/TX_E_HF	1				Bizarre Emulator FIFO Flag
/TX_E_EF	1				Another Bizarre Emulator FIFO Flag
<b>5</b>					
Miscellaneous					Miscellaneous
/TSI_R	1				TSI Emulator mode indicator
/TSI_G	1				"
/XQT	1				Global Execute Line
BXING	1				Bunch Crossing
VBX	1				Virtual Bunch Crossing
/CONFIG4	1				Configuration Command
EXT_TRIGGER	1				External Trigger
U_DOIT[1:0]		2			Taxi_dec Control Lines

RF_OUTA	1			FIFO Clock
RSM_CLKA	1			State Machine Clock
<b>11</b>				
<b>Status</b>				<b>Status</b>
/DONE	1			Global Done
/L1_DONE	1			Level One Done
/WAIT	1			Global Wait
TXVLTN	1			Taxi Violation
/ERROR	1			Global Error
<b>5</b>				
<b>SRC Lines</b>				<b>SRC Lines</b>
/TEST		1		
/RUN		1		
/RECOVER		1		
/HALT		1		
/CONTROL		1		
/CAL		1		
CALIB[2:0]		3		
EID[3:0]		4		
/L1A		1		
L1A_ADDR[1:0]		2		
/L2A		1		
L2A_ADDR[1:0]		2		
/L2R		1		
L2R_ADDR[1:0]		2		
<b>22</b>				
<b>TOTAL</b>				
	<b>68</b>			

## 8.2 Constraint File

```
# Design: taxi20
# Created by XACT Floorplanner ver 6.0.0
# Actually created by JO by hand on March 20, 1997
# and resides in Ver1_4, Rev 2
NOTPLACE BLOCK *: PAD46 PAD132 PAD111;

# place block BXING      : P16 ;
place block -CAL         : C9 ;
place block -CALI        : R6 ;
place block CALIB0       : A10 ;
place block CALIB1       : A11 ;
place block CALIB2       : E16 ;
place block -CONTROL     : D6 ;
place block EID0         : F14 ;
place block EID1         : D16 ;
place block EID2         : E15 ;
place block EID3         : E14 ;
place block -EN_RSM      : D13 ;
place block EXT_TRIG     : N10 ;
place block -HALT        : C1 ;
place block -L1A         : F15 ;
place block L1A_ADD0     : G14 ;
place block L1A_ADD1     : F16 ;
place block -L2A         : F1 ;
place block L2A_ADD0     : E1 ;
place block L2A_ADD1     : E2 ;
place block -L2R         : D2 ;
place block L2R_ADD0     : F2 ;
place block L2R_ADD1     : F3 ;
place block -BWRITE       : H2 ;
place block -CONFIG4     : C2 ;
place block -DONE         : H1 ;
place block -ERROR        : N1 ;
place block -L1_DONE      : K16 ;
place block QRD0          : J15 ;
place block QRD1          : G16 ;
# place block -RD_TX_E    : G3 ;
# place block -RES_TX_E   : P6 ;
place block -STROBE0      : A6 ;
place block -STROBE1      : A14 ;
place block -TAXI         : K2 ;
# place block -TX_E_EF    : P10 ;
# place block -TX_E_HF    : R9 ;
place block -VMEOK4       : M1 ;
place block -WAIT         : G2 ;
# place block -WRT_TX_E   : L2 ;
place block -RECOVER       : D1 ;
place block -RUN           : A8 ;
place block -TEST          : B7 ;
# place block TXVLTN      : R8 ;
place block -TSI_R         : B6 ;
```

```
place block -TSI_G : B9 ;
place block U_DOIT0 : C5 ;
place block U_DOIT1 : C7 ;
place block VADDR1 : K1 ;
place block VADDR2 : J2 ;
place block VADDR3 : L1 ;
place block VADDR4 : K3 ;
place block VBX : N15 ;
place block VDAT0 : T8 ;
place block VDAT1 : T7 ;
place block VDAT2 : N7 ;
place block VDAT3 : P7 ;
place block VDAT4 : P9 ;
place block VDAT5 : P8 ;
place block VDAT6 : T10 ;
place block VDAT7 : T9 ;
place block VDAT8 : P11 ;
place block VDAT9 : T12 ;
place block VDAT10 : T13 ;
place block VDAT11 : N12 ;
place block VDAT12 : R13 ;
place block VDAT13 : P13 ;
place block VDAT14 : R12 ;
place block VDAT15 : P12 ;
place block -XQT : T5 ;
# End
```

### ***8.3 Placement Report***

Error! Not a valid filename.

#### ***8.4 XACT Performance***

Error! Not a valid filename.

## 8.5 State CAD Diagrams

### 8.5.1 VME

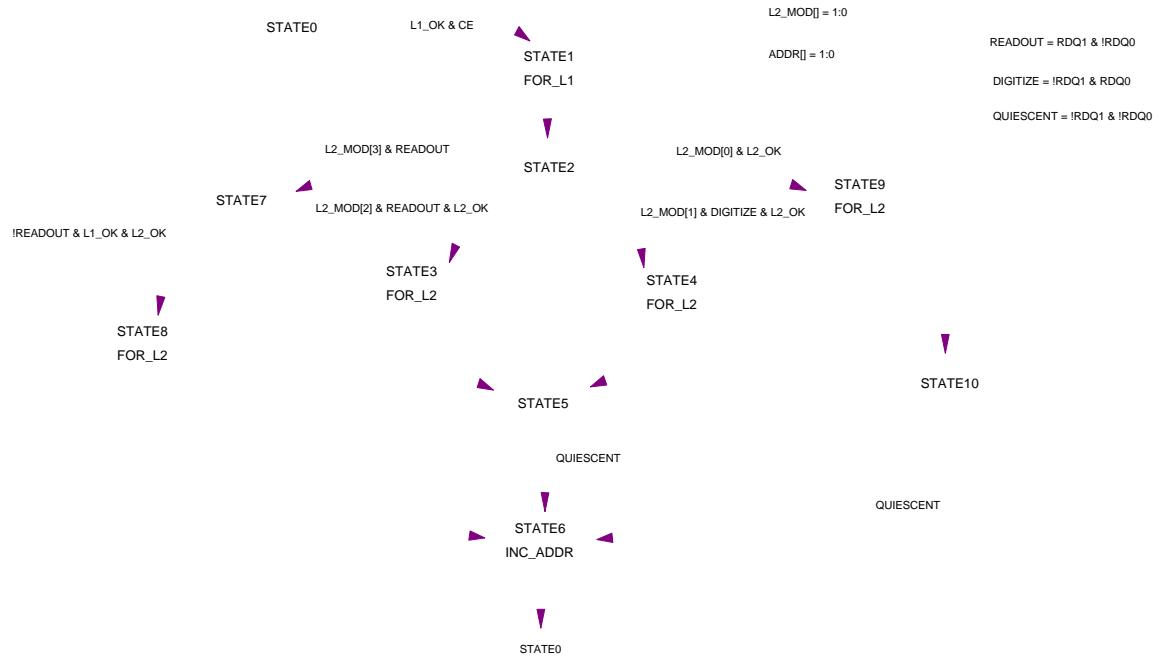
```
/cdf/src/tsi_em/vme.dia
Taxi emulator vme decoder
March 24, 1997

add[ ] = 3:0

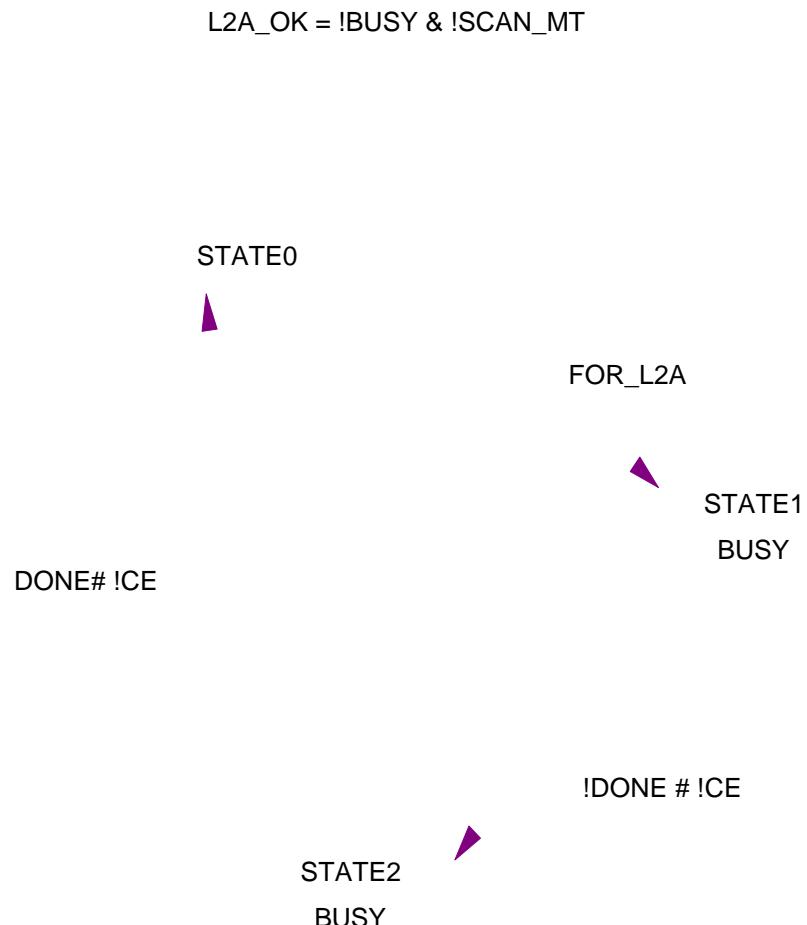
reg0 = add[0] & taxi
reg1 = add[1] & taxi
reg4 = add[4] & taxi
reg5 = add[5] & taxi
reg6 = add[6] & taxi
reg7 = add[7] & taxi
reg8 = add[8] & taxi
reg9 = add[9] & taxi
reg10 = add[10] & taxi
reg11 = add[11] & taxi

vmeok = taxi & (add[0] or add[1] or add[4] or add[5] or add[6] or add[7] or add[9] or add[10] c
chip = taxi & write
pad = taxi & !write
```

### 8.5.2 AUTO2



### 8.5.3 L2A\_ST

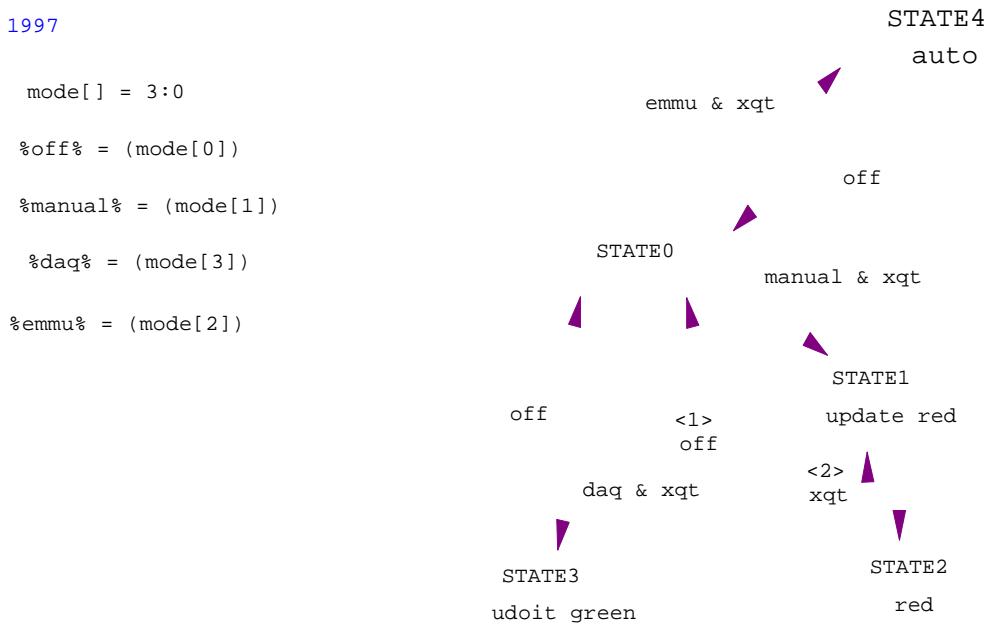


#### 8.5.4 MODES

```
/cdf/src/tsi_em/modes.dia

Indicators : RED is emulation mode
              GREEN is normal daq
              ORANGE is external trigger
              OFF is off
```

Mar 24, 1997



## **8.6 HDL Code**

### **8.6.1 VME**

```
" D:\CDF\SRC\TSI_EM\VME.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.10
" Sun Jun 22 14:36:56 1997

" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are speed optimized.
```

MODULE VME

#### **DECLARATIONS**

"Input variables

```
add0 PIN;
add1 PIN;
add2 PIN;
add3 PIN;
taxi PIN;
write PIN;
```

"Logic variables

```
chip PIN ISTYPE 'com';
pad PIN ISTYPE 'com';
reg0 PIN ISTYPE 'com';
reg1 PIN ISTYPE 'com';
reg4 PIN ISTYPE 'com';
reg5 PIN ISTYPE 'com';
reg6 PIN ISTYPE 'com';
reg7 PIN ISTYPE 'com';
reg8 PIN ISTYPE 'com';
reg9 PIN ISTYPE 'com';
reg10 PIN ISTYPE 'com';
reg11 PIN ISTYPE 'com';
vmeok PIN ISTYPE 'com';
```

"Vectors

#### **DECLARATIONS**

```
add=[  
      add3,  
      add2,  
      add1,  
      add0  
    ];
```

"Logic Equations

EQUATIONS

```
chip = taxi & write ;  
  
pad = taxi & !write ;  
  
reg0 = taxi & !add3 & !add2 & !add1 & !add0 ;  
  
reg1 = taxi & !add3 & !add2 & !add1 & add0 ;  
  
reg4 = taxi & !add3 & add2 & !add1 & !add0 ;  
  
reg5 = taxi & !add3 & add2 & !add1 & add0 ;  
  
reg6 = taxi & !add3 & add2 & add1 & !add0 ;  
  
reg7 = taxi & !add3 & add2 & add1 & add0 ;  
  
reg8 = taxi & add3 & !add2 & !add1 & !add0 ;  
  
reg9 = taxi & add3 & !add2 & !add1 & add0 ;  
  
reg10 = taxi & add3 & !add2 & add1 & !add0 ;  
  
reg11 = taxi & add3 & !add2 & add1 & add0 ;  
  
vmeok = add3 & !add2 & add1 & taxi # add3 & !add2 & add0 & taxi # !add3  
      & add2 & taxi # !add2 & !add1 & add0 & taxi # !add3 & !add1 & !add0 &  
      taxi ;
```

END VME

### 8.6.2 AUTO2

```
" E:\HEPL\CDF\SRC\TSI_EM\AUTO2.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.10
" Fri Jun 13 13:08:21 1997

" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are area optimized.

MODULE AUTO2

DECLARATIONS
"clock name
    CLK PIN;

"Input variables
    ADDR0 PIN;
    ADDR1 PIN;
    CE PIN;
    L1_OK PIN;
    L2_MODO PIN;
    L2_MOD1 PIN;
    L2_OK PIN;
    RDQ0 PIN;
    RDQ1 PIN;

"Output variables
    FOR_L1 PIN ISTYPE 'com';
    FOR_L2 PIN ISTYPE 'com';
    INC_ADDR PIN ISTYPE 'com';

"Logic variables
    DIGITIZE PIN ISTYPE 'com';
    QUIESCENT PIN ISTYPE 'com';
    READOUT PIN ISTYPE 'com';

"State variables
    SV0 PIN ISTYPE 'reg';
    SV1 PIN ISTYPE 'reg';
    SV2 PIN ISTYPE 'reg';
    SV3 PIN ISTYPE 'reg';

"Vectors
DECLARATIONS
    ADDR=[  
        ADDR1,  
        ADDR0  
    ];  
  
    L2_MOD=[
```

```

        L2_MOD1,
        L2_MOD0
    ];

"State Register assignment
DECLARATIONS
    sreg=[ SV0,SV1,SV2,SV3];

EQUATIONS
    sreg.clk=CLK;

DECLARATIONS
    STATE0=[0, 0, 0, 0];
    STATE1=[0, 0, 0, 1];
    STATE2=[0, 0, 1, 0];
    STATE3=[0, 0, 1, 1];
    STATE4=[0, 1, 0, 0];
    STATE5=[0, 1, 0, 1];
    STATE6=[0, 1, 1, 0];
    STATE7=[0, 1, 1, 1];
    STATE8=[1, 0, 0, 0];
    STATE9=[1, 0, 0, 1];
    STATE10=[1, 0, 1, 0];

EQUATIONS

SV0 := !READOUT & L1_OK & L2_OK & !SV0.FB & SV1.FB & SV2.FB & SV3.FB # 
    L2_OK & !L2_MOD1 & !L2_MOD0 & !SV0.FB & !SV1.FB & SV2.FB & !SV3.FB # SV0.FB
    & !SV1.FB & !SV2.FB & SV3.FB # !QUIESCENT & SV0.FB & !SV1.FB & SV2.FB & !
    SV3.FB ;

SV1 := DIGITIZE & L2_OK & !L2_MOD1 & L2_MOD0 & !SV0.FB & !SV1.FB & SV2.FB # 
    !SV0.FB & !SV1.FB & SV2.FB & SV3.FB # !SV0.FB & SV1.FB & !SV2.FB #
    SV0.FB & !SV1.FB & !SV2.FB & !SV3.FB # QUIESCENT & SV0.FB & !SV1.FB & !
    SV3.FB # READOUT & L2_MOD1 & L2_MOD0 & !SV0.FB & !SV1.FB & SV2.FB #
    READOUT & !SV0.FB & SV2.FB & SV3.FB # !L1_OK & !SV0.FB & SV2.FB & SV3.FB #
    !L2_OK & !SV0.FB & SV2.FB & SV3.FB ;

SV2 := L2_MOD0 & !DIGITIZE & !SV1.FB & SV2.FB & !SV3.FB # !L2_OK & !SV1.FB
    & SV2.FB & !SV3.FB # QUIESCENT & !SV0.FB & !SV2.FB & SV3.FB # SV0.FB & !
    SV1.FB & !SV2.FB # L2_MOD1 & !SV1.FB & SV2.FB & !SV3.FB # READOUT & !
    SV0.FB & SV1.FB & SV2.FB & SV3.FB # !L1_OK & !SV0.FB & SV1.FB & SV2.FB &
    SV3.FB # !L2_OK & !SV0.FB & SV1.FB & SV2.FB & SV3.FB # !SV1.FB & !SV2.FB
    & SV3.FB # SV0.FB & !SV1.FB & !SV3.FB ;

SV3 := L1_OK & CE & !SV0.FB & !SV2.FB & !SV3.FB # READOUT & L2_OK & L2_MOD1
    & !SV0.FB & !SV1.FB & SV2.FB # !SV0.FB & !SV1.FB & SV2.FB & SV3.FB # !
    SV0.FB & SV1.FB & !SV2.FB & !SV3.FB # !QUIESCENT & !SV0.FB & SV1.FB & !
    SV2.FB # READOUT & L2_MOD1 & L2_MOD0 & !SV0.FB & !SV1.FB & SV2.FB #
    READOUT & !SV0.FB & SV2.FB & SV3.FB # !L1_OK & !SV0.FB & SV2.FB & SV3.FB #
    !L2_OK & !SV0.FB & SV2.FB & SV3.FB # L2_OK & !L2_MOD1 & !L2_MOD0 & !
    SV0.FB & !SV1.FB & SV2.FB ;

FOR_L1= SV3.FB & !SV2.FB & !SV1.FB & !SV0.FB ;

```

```
FOR_L2= SV3.FB & SV2.FB & !SV1.FB & !SV0.FB # !SV3.FB & !SV2.FB & SV1.FB &
!SV0.FB # !SV2.FB & !SV1.FB & SV0.FB ;

INC_ADDR= !SV3.FB & SV2.FB & SV1.FB & !SV0.FB ;
"Logic Equations
EQUATIONS
DIGITIZE = !RDQ1 & RDQ0 ;

QUIESCENT = !RDQ1 & !RDQ0 ;

READOUT = RDQ1 & !RDQ0 ;

END AUTO2
```

### 8.6.3 L2A\_ST

```
" D:\CDF\SRC\TSI_EM\L2A_ST.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.10
" Tue Jul 01 10:56:51 1997

" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are area optimized.

MODULE L2A_ST

DECLARATIONS
"clock name
    CLK PIN;

"Input variables
    CE PIN;
    DONE PIN;
    FOR_L2A PIN;
    SCAN_MT PIN;

"Output variables
    BUSY PIN ISTYPE 'com';

"Logic variables
    L2A_OK PIN ISTYPE 'com';

"State variables
    SV0 PIN ISTYPE 'reg';
    SV1 PIN ISTYPE 'reg';

"State Register assignment
DECLARATIONS
    sreg=[ SV0,SV1];

EQUATIONS
    sreg.clk=CLK;

DECLARATIONS
    STATE0=[0, 0];
    STATE1=[0, 1];
    STATE2=[1, 0];

EQUATIONS
    SV0 := !CE & !SV0.FB & SV1.FB # !DONE & !SV0.FB & SV1.FB # !DONE & CE &
        SV0.FB & !SV1.FB ;
    SV1 := FOR_L2A & !SV0.FB & !SV1.FB # DONE & CE & !SV0.FB & SV1.FB ;
```

```
BUSY= SV1.FB & !SV0.FB # !SV1.FB & SV0.FB ;
"Logic Equations
EQUATIONS
L2A_OK = !BUSY & !SCAN_MT ;

END L2A_ST
```

#### 8.6.4 MODES

```
" D:\CDF\SRC\TSI_EM\MODES.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.10
" Fri Aug 29 12:53:07 1997

" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are speed optimized.
```

#### MODULE MODES

##### DECLARATIONS

```
"clock name
    CLK PIN;

"Input variables
    CREG0 PIN;
    CREG1 PIN;
    CREG2 PIN;
    CREG3 PIN;
    CREG4 PIN;
    CREG5 PIN;
    CREG6 PIN;
    CREG7 PIN;
    xqt PIN;
```

##### "Output variables

```
    green PIN ISTYPE 'reg';
    red PIN ISTYPE 'reg';
    udoit PIN ISTYPE 'reg';
    update PIN ISTYPE 'reg';
```

##### "Logic variables

```
    auto PIN ISTYPE 'com';
```

##### "State variables

```
    SV0 PIN ISTYPE 'reg';
    SV1 PIN ISTYPE 'reg';
    SV2 PIN ISTYPE 'reg';
```

##### "Vectors

##### DECLARATIONS

```
mode=[
    CREG7,
    CREG6,
    CREG5,
    CREG4
];
```

```
SM=[
```

```

CREG3,
CREG2,
CREG1,
CREG0
];

"E clocked logic clock setup
EQUATIONS
    green.clk=CLK;
    red.clk=CLK;
    udoit.clk=CLK;
    update.clk=CLK;

"State Register assignment
DECLARATIONS
    sreg=[ SV0,SV1,SV2];

EQUATIONS
    sreg.clk=CLK;

DECLARATIONS
    STATE0=[0, 0, 0];
    STATE1=[0, 0, 1];
    STATE2=[0, 1, 0];
    STATE3=[0, 1, 1];
    STATE4=[1, 0, 0];
    STATE6=[1, 0, 1];

EQUATIONS
    SV0 := xqt & CREG3 & !CREG2 & !CREG1 & !CREG0 & !CREG7 & !CREG6 & CREG5 & !
        CREG4 & !SV1.FB & !SV2.FB # SV0.FB & !SV1.FB & !SV2.FB # CREG6 & SV0.FB &
        !SV1.FB # CREG4 & SV0.FB & !SV1.FB # CREG7 & SV0.FB & !SV1.FB # CREG5
        & SV0.FB & !SV1.FB ;

    SV1 := !SV0.FB & !SV1.FB & SV2.FB # CREG6 & !xqt & !SV0.FB & SV1.FB #
        CREG4 & !xqt & !SV0.FB & SV1.FB # CREG7 & !xqt & !SV0.FB & SV1.FB # CREG5
        & !xqt & !SV0.FB & SV1.FB # xqt & !CREG7 & !CREG6 & CREG5 & CREG4 & !
        SV0.FB & !SV1.FB # CREG7 & !SV0.FB & SV2.FB # CREG6 & !SV0.FB & SV2.FB #
        CREG5 & !SV0.FB & SV2.FB # CREG4 & !SV0.FB & SV2.FB ;

    SV2 := xqt & !CREG3 & !CREG7 & !CREG6 & CREG5 & !SV0.FB & !SV2.FB # xqt &
        CREG2 & !CREG7 & !CREG6 & CREG5 & !SV0.FB & !SV2.FB # xqt & CREG1 & !CREG7
        & !CREG6 & CREG5 & !SV0.FB & !SV2.FB # xqt & CREG0 & !CREG7 & !CREG6 &
        CREG5 & !SV0.FB & !SV2.FB # CREG4 & xqt & !SV0.FB & SV1.FB # CREG5 & xqt
        & !SV0.FB & SV1.FB # CREG6 & xqt & !SV0.FB & SV1.FB # CREG7 & xqt & !
        SV0.FB & SV1.FB # xqt & !CREG7 & !CREG6 & CREG4 & !SV0.FB & !SV2.FB #
        CREG7 & !SV0.FB & SV1.FB & SV2.FB # CREG6 & !SV0.FB & SV1.FB & SV2.FB #
        CREG5 & !SV0.FB & SV1.FB & SV2.FB # CREG4 & !SV0.FB & SV1.FB & SV2.FB #
        SV0.FB & !SV1.FB & !SV2.FB # CREG6 & !xqt & SV0.FB & !SV1.FB # CREG4 & !
        xqt & SV0.FB & !SV1.FB # CREG7 & !xqt & SV0.FB & !SV1.FB # CREG5 & !xqt &
        SV0.FB & !SV1.FB ;

```

```

green := xqt & !CREG7 & !CREG6 & CREG5 & CREG4 & !SV1.FB & !SV2.FB # CREG7
& !SV0.FB & SV1.FB & SV2.FB # CREG6 & !SV0.FB & SV1.FB & SV2.FB # CREG5 &
!SV0.FB & SV1.FB & SV2.FB # CREG4 & !SV0.FB & SV1.FB & SV2.FB # xqt &
CREG3 & !CREG2 & !CREG1 & !CREG0 & !CREG7 & !CREG6 & CREG5 & !SV1.FB & !
SV2.FB # SV0.FB & !SV1.FB & !SV2.FB # CREG6 & SV0.FB & !SV1.FB # CREG4
& SV0.FB & !SV1.FB # CREG7 & SV0.FB & !SV1.FB # CREG5 & SV0.FB & !SV1.FB
;

red := xqt & !CREG7 & !CREG6 & !CREG5 & CREG4 & !SV0.FB & !SV2.FB # xqt & !
CREG3 & !CREG7 & !CREG6 & CREG5 & !CREG4 & !SV0.FB & !SV2.FB # xqt & CREG2
& !CREG7 & !CREG6 & CREG5 & !CREG4 & !SV0.FB & !SV2.FB # xqt & CREG1 & !
CREG7 & !CREG6 & CREG5 & !CREG4 & !SV0.FB & !SV2.FB # xqt & CREG0 & !CREG7
& !CREG6 & CREG5 & !CREG4 & !SV0.FB & !SV2.FB # !SV0.FB & !SV1.FB & SV2.FB
# CREG6 & !SV0.FB & SV1.FB & !SV2.FB # CREG4 & !SV0.FB & SV1.FB & !SV2.FB
# CREG7 & !SV0.FB & SV1.FB & !SV2.FB # CREG5 & !SV0.FB & SV1.FB & !
SV2.FB # xqt & !CREG7 & !CREG6 & CREG5 & !CREG4 & !SV0.FB & !SV1.FB #
SV0.FB & !SV1.FB & !SV2.FB # CREG6 & !SV1.FB & SV2.FB # CREG4 & !SV1.FB &
SV2.FB # CREG7 & !SV1.FB & SV2.FB # CREG5 & !SV1.FB & SV2.FB ;

udoit := xqt & !CREG7 & !CREG6 & CREG5 & CREG4 & !SV0.FB & !SV1.FB & !SV2.FB
# CREG7 & !SV0.FB & SV1.FB & SV2.FB # CREG6 & !SV0.FB & SV1.FB & SV2.FB
# CREG5 & !SV0.FB & SV1.FB & SV2.FB # CREG4 & !SV0.FB & SV1.FB & SV2.FB ;

update := xqt & !CREG7 & !CREG6 & !CREG5 & CREG4 & !SV0.FB & !SV2.FB #
CREG4 & xqt & !SV0.FB & SV1.FB & !SV2.FB # CREG5 & xqt & !SV0.FB & SV1.FB &
!SV2.FB # CREG6 & xqt & !SV0.FB & SV1.FB & !SV2.FB # CREG7 & xqt & !
SV0.FB & SV1.FB & !SV2.FB # xqt & !CREG7 & !CREG6 & CREG5 & !CREG4 & !
SV0.FB & !SV2.FB # CREG4 & xqt & !SV0.FB & !SV1.FB & SV2.FB # CREG5 & xqt
& !SV0.FB & !SV1.FB & SV2.FB # CREG6 & xqt & !SV0.FB & !SV1.FB & SV2.FB #
CREG7 & xqt & !SV0.FB & !SV1.FB & SV2.FB ;

"Logic Equations
EQUATIONS
auto = red.FB & !CREG7 & !CREG6 & CREG5 & !CREG4 ;

END MODES

```

## 9. Pipe Cap

### 9.1 Pin Description

Design					
Date					PIPE-CAP
Signal	I	O	OT	I/O	Function
/PIPE_CAP	1				VME Request
/STROBE0	1				VME Strobe
/BWRITE	1				VME Write line
VADDR[3:1]	3				VME Address bus
VDAT[7:0]				8	VME Data bus
/VMEOK5		1			VME local acknowledge
/XQT	1				Board level Execute
/L1A	1				Level-1 Accept
/L1_DONE	1				Release Pipe Cap
/RUN	1				CDF Global RUN mode
RSM_CLK	1				State machine clock
/CONFIG5		1			Configuration indicator
/BAR_GRAPH[3:0]		4			Bar graph L1 occupancy indicators
PIPE_CAP[5:0]		6			Pipe Cap emulation number
<b>Pin Count</b>	<b>11</b>	<b>12</b>	<b>0</b>	<b>8</b>	<b>31</b>

## 9.2 Constraint File

```
# Initial constraint file for PIPE_CAP just to nail down the pins
# Nov. 1, 1996
# Target device xc4010epc84
notplace instance *: p41 p71;
place instance -BWRITE_pad      : P16 ;
place instance -CONFIG2_pad     : P37;
place instance -L1_DONE_pad     : P81 ;
place instance -L1A_pad        : P80 ;
place instance -PIPE_CAP_pad   : P14 ;
place instance N_RUN_pad       : P82 ;
place instance -STROBE0_pad    : P13 ;
place instance N_XQT_pad       : P79 ;
place instance PIPE_CAP0_pad   : P65 ;
place instance PIPE_CAP1_pad   : P62 ;
place instance PIPE_CAP2_pad   : P61 ;
place instance PIPE_CAP3_pad   : P60 ;
place instance PIPE_CAP4_pad   : P59 ;
place instance PIPE_CAP5_pad   : P58 ;
place instance RSM_CLK_pad     : P35 ;
place instance VADDR1_pad      : P19 ;
place instance VADDR2_pad      : P18 ;
place instance VADDR3_pad      : P17 ;
place instance VDAT0_pad       : P36 ;
place instance VDAT1_pad       : P28 ;
place instance VDAT2_pad       : P27 ;
place instance VDAT3_pad       : P26 ;
place instance VDAT4_pad       : P25 ;
place instance VDAT5_pad       : P24 ;
place instance VDAT6_pad       : P23 ;
place instance VDAT7_pad       : P20 ;
place instance -RES_PIPE_pad   : P49 ;
place instance -VMEOK5_pad     : P29 ;

place instance L1A_AX0          : P38 ;
place instance L1A_AX1          : P39 ;
place instance L1A_AX2          : P40 ;
place instance L1A_AX3          : P44 ;
place instance L1A_AX4          : P9 ;
place instance L1A_AX5          : P8 ;

place instance WR_AX0           : P7 ;
place instance WR_AX1           : P6 ;
place instance WR_AX2           : P5 ;
place instance WR_AX3           : P48 ;
place instance WR_AX4           : P47 ;
place instance WR_AX5           : P46 ;
```

### **9.3 Placement Report**

PLACEMENT RESULTS FOR DESIGN TEST2  
From PPR Version 5.2.1

1997/03/05 18:31:40

Xilinx, Inc.  
(c) Copyright 1997. All Rights Reserved.

#### Report Contents

---

1. List of Warnings
2. Device Utilization
3. Implementation Options

#### List of Warnings

---

\*\*\* PPR: WARNING 7034:

In categories of paths that have NO timing specifications, the design has 1566 path end-point pairs. Note that there may be multiple paths between any listed pair of end-points. To limit the number of paths reported in each category, set PPR parameter show\_tsi\_paths = <value>.

\*\*\* PPR: WARNING 5527:

The design uses no global buffers (BUFGS or BUFGP). Their use can improve performance for clocks or widely distributed signals.

#### Device Utilization

---

#### Partitioned Design Utilization Using Part 4010EPC84-4

---

	No. Used	Max Available	%Used
Occupied CLBs	287	400	71%
Bonded I/O Pins	18	61	29%
F and G Function Generators (*)	458	800	57%
H Function Generators	171	400	42%
CLB Flip Flops	77	800	9%
IOB Input Flip Flops	0	160	0%
IOB Output Flip Flops	0	160	0%

3-State Buffers	0	880	0%
3-State Half Longlines	0	80	0%
Edge Decode Inputs	0	240	0%
Edge Decode Half Longlines	0	32	0%
CLB Fast Carry Logic	8	400	2%

(\*) If RAM/ROM elements are present in the design, this count includes the function generators used for them. A 16x1 memory uses 1 function generator; a 32x1 uses two.

---

#### CPU Times

CPU time taken for Partition: 0 hrs 0 mins 19 secs  
 CPU time taken for Placement: 0 hrs 4 mins 39 secs

#### Implementation Options

---

#### PPR Parameters

Design	= test2
Parttype	= from design file
LogFile	= ppr.log
Outfile	= <design name>
Estimate	= FALSE

#### Additional Specified or Non-Default Parameters

paramfile	= params.txt
seed	= 857586378
design	= test2
run_pic2map	= true
placer_effort	= 2
router_effort	= 2
path_timing	= true
ignore_cstfile	= true
route_thru_bufg	= ok
route_thru_blk	= ok
guide_blk	= all
lock_routing	= whole_sigs
split_report	= true

#### Parameter Values from XACTINIT.DAT

ORCAD_NAMES	= false
-------------	---------

===== End of Report =====



## 9.4 XACT Performance

XACT PERFORMANCE RESULTS FOR DESIGN TEST2  
From PPR Version 5.2.1

1997/03/05 18:39:41

Xilinx, Inc.

(c) Copyright 1997. All Rights Reserved.

Xact Performance Summary

---

Parttype Used : 4010EPC84

Speed Grade : -4

End-

Limit	Actual	Points
(ns)	*	(ns) Missed Specification

---

<auto>	99.6	0/71	DEFAULT_FROM_FFS_TO_FFS=FROM:ffs:TO:ffs
<auto>	89.1	0/216	DEFAULT_FROM_PADS_TO_FFS=FROM:pads:TO:ffs
<auto>	26.3	0/6	DEFAULT_FROM_FFS_TO_PADS=FROM:ffs:TO:pads

(\*) Use the -FailedSpec and/or -TSMaxPaths options of the XDelay-TimeSpec command, accessible through the XDE or XDelay program, to confirm the actual path delays computed by PPR. Note that XDelay-TimeSpec will not report paths that start and end in the same block (CLB or IOB) and use no external routing.

\*\*\* PPR: WARNING 7028:

The design has flip-flops with asynchronous set/reset controls (PRE/SD or CLR/RD pins). When PPR analyzes design timing, it does not trace paths through the asynchronous set/reset input and on through the Q output.

If you want PPR to control the delay on paths through asynchronous set/reset pins, you must split the delay requirement into two segments: one ending at the set/reset input, and the other beginning at the flip-flop output. If you want PPR not to analyze paths that lead to asynchronous set/reset pins, attach an IGNORE specification to the pin(s) or signal(s).

By default, XDelay reports all paths through asynchronous set/reset pins. To prevent XDelay from showing these paths, use FlagBlk CLB\_Disable\_SR\_Q on the appropriate flip-flops.

\*\*\* PPR: WARNING 7029:

The design has RAM elements. When PPR analyzes design timing, it does not trace paths through the data or write-enable inputs and on through the RAM output. Such paths are normally of interest only when the RAM

is being read during a write operation.

If you want PPR to control the delay on paths through the D or WE inputs, you must split the delay requirement into two segments: one ending at the RAM input pin, and the other beginning at the RAM output.

By default, XDelay does not report paths through D or WE pins either. To allow XDelay to show these paths, use FlagBlk CLB\_Enable\_WE and/or FlagBlk CLB\_Enable\_DIN on the appropriate RAMs.

===== End of Report =====

## 9.5 State CAD Diagrams

### 9.5.1 PIPE\_VME

\cdf\src\pipe\_cap\pipe\_vme  
VME Interface  
6/2/97 NF

ADDR[] = 5:0

%R0% = ADDR[<sup>h</sup>0]

CONTROL register: MODE, SUB-MODE (R/W)

%R2% = ADDR[<sup>h</sup>2]

Pipe length

REG0 = R0 & REQ

REG2 = R2 & REQ

DATA\_OUT = VMEOK & !WRITE

DATA\_IN = VMEOK & WRITE

VMEOK = REQ & (R0 or R2)

## 9.6 HDL Code

### 9.6.1 PIPE\_VME

```
" E:\HEPL\CDF\SRC\PIPE_CAP\PIPE_VME.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.10
" Wed Jun 18 14:58:26 1997
```

```
" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are area optimized.
```

```
MODULE PIPE_VME
```

```
DECLARATIONS
```

```
"Input variables
```

```
    ADDR0 PIN;
    ADDR1 PIN;
    ADDR2 PIN;
    ADDR3 PIN;
    ADDR4 PIN;
    ADDR5 PIN;
    REQ PIN;
    WRITE PIN;
```

```
"Logic variables
```

```
    DATA_IN PIN ISTYPE 'com';
    DATA_OUT PIN ISTYPE 'com';
    REG0 PIN ISTYPE 'com';
    REG2 PIN ISTYPE 'com';
    VMEOK PIN ISTYPE 'com';
```

```
"Vectors
```

```
DECLARATIONS
```

```
    ADDR=[  
        ADDR5,  
        ADDR4,  
        ADDR3,  
        ADDR2,  
        ADDR1,  
        ADDR0  
    ];
```

```
"Logic Equations
```

```
EQUATIONS
```

```
DATA_IN = VMEOK & WRITE ;
```

```
DATA_OUT = VMEOK & !WRITE ;
```

```
REG0 = REQ & !ADDR5 & !ADDR4 & !ADDR3 & !ADDR2 & !ADDR1 & !ADDR0 ;  
REG2 = REQ & !ADDR5 & !ADDR4 & !ADDR3 & !ADDR2 & ADDR1 & !ADDR0 ;  
VMEOK = !ADDR5 & !ADDR4 & !ADDR3 & !ADDR2 & !ADDR0 & REQ ;  
END PIPE_VME
```

## 10. Error Logger

### 10.1 Pin Description

Design					CDF/SRC/ERR_LOG
Date					10/2/96
					3/20/98 14:16
Signal	I	O	OT	I/O	Function
					Error Inputs
/RSM_ERR[5:0]	6				SRC generated error flags
/GLFE	1				G-Link fatal error
/GLNFE	1				G-Link non-fatal error
VRB_STS[9:2]	8				VRB generated error flags
<b>16</b>					
					Error fifos
VRB_ERR[7:0]		8			VRB error outputs to error fifo
SRC_ERR[7:0]		8			SRC error outputs to fifo
/RD_EF[4:0]		5			Read Error fifo
/WRT_EF[4:0]		5			Write Error fifos
/RES_EF[4:0]		5			Reset Error fifos
/EF_EF[4:0]	5				Error fifo full flag
/PAFE_EF[4:0]	5				Error fifo Almost Empty/Full
/HF_EF[4:0]	5				Error fifo Empty flags
/EN_EF2VME[4:0]		5			Enable Error fifos to VME bus
<b>51</b>					
					VME Access
/ERR_LOG	1				VME Request line for ERR_LOG FPGA
/BWRITE	1				VME Board write line
/STROBE[1:0]	2				VME data strobes
/VMEOK5		1			Local acknowledge
VDAT[15:0]			16		VME data (2 bytes)
VADDR[5:1]	5				32 word address space
<b>26</b>					
					Misc
/XQT	1				Global Execute line
/RUN	1				RUN command from TSI
RSM_CLK	1				132 ns input clock
RF5	1				53 MHz clock
/CONFIG5		1			Local configuration indicator
<b>5</b>					
Pin count	44	38	0	16	<b>98</b>
98					

## 10.2 Constraint File

```
# Design: Err_log
# Created by XACT Floorplanner ver 6.0.1
NOTPLACE BLOCK *: R3 C15 H15;
PLACE BLOCK -CONFIG5: P15;

PLACE BLOCK N_RD4: VQ;
PLACE BLOCK N_RD2: VP;
PLACE BLOCK N_RD0: VO;
PLACE BLOCK N_EN2VME4: UQ;
PLACE BLOCK N_EN2VME2: UP;
PLACE BLOCK N_EN2VME1: UO;
PLACE BLOCK VDAT13: P8;
PLACE BLOCK VDAT12: R8;
PLACE BLOCK VDAT11: R9;
PLACE BLOCK VDAT10: P9;
PLACE BLOCK VDAT9: T10;
PLACE BLOCK VDAT8: T9;
PLACE BLOCK VDAT7: P10;
PLACE BLOCK VDAT6: N10;
PLACE BLOCK VDAT5: T11;
PLACE BLOCK VDAT4: R10;
PLACE BLOCK VDAT3: N11;
PLACE BLOCK VDAT2: R11;
PLACE BLOCK VDAT1: T12;
PLACE BLOCK VDAT0: P11;
PLACE BLOCK COUNTER7/T12: DQ;
PLACE BLOCK COUNTER7/T11: EQ;
PLACE BLOCK COUNTER7/T8: FQ;
PLACE BLOCK COUNTER7/T7: GQ;
PLACE BLOCK COUNTER7/T4: HQ;
PLACE BLOCK COUNTER7/Q10: DR;
PLACE BLOCK COUNTER7/Q8: ER;
PLACE BLOCK COUNTER7/Q6: FR;
PLACE BLOCK COUNTER7/Q4: GR;
PLACE BLOCK COUNTER7/Q2: HR;
PLACE BLOCK COUNTER7/Q0: IR;
PLACE BLOCK COUNTER7/Q12: CR;
PLACE BLOCK COUNTER7/$1I160: TBUF.DS.1;
PLACE BLOCK COUNTER7/$1I156: TBUF.CS.2;
PLACE BLOCK COUNTER7/$1I152: TBUF.ES.1;
PLACE BLOCK COUNTER7/$1I148: TBUF.DS.2;
PLACE BLOCK COUNTER7/$1I144: TBUF.FS.1;
PLACE BLOCK COUNTER7/$1I140: TBUF.ES.2;
PLACE BLOCK COUNTER7/$1I136: TBUF.GS.1;
PLACE BLOCK COUNTER7/$1I132: TBUF.FS.2;
PLACE BLOCK COUNTER7/$1I128: TBUF.HS.1;
PLACE BLOCK COUNTER7/$1I124: TBUF.GS.2;
PLACE BLOCK COUNTER7/$1I120: TBUF.IS.1;
PLACE BLOCK COUNTER7/$1I116: TBUF.HS.2;
PLACE BLOCK COUNTER7/$1I112: TBUF.JS.1;
```

PLACE BLOCK COUNTER7/\$1I70: TBUF.IS.2;  
PLACE BLOCK COUNTER6/T12: DO;  
PLACE BLOCK COUNTER6/T11: EO;  
PLACE BLOCK COUNTER6/T8: FO;  
PLACE BLOCK COUNTER6/T7: GO;  
PLACE BLOCK COUNTER6/T4: HO;  
PLACE BLOCK COUNTER6/Q10: DP;  
PLACE BLOCK COUNTER6/Q8: EP;  
PLACE BLOCK COUNTER6/Q6: FP;  
PLACE BLOCK COUNTER6/Q4: GP;  
PLACE BLOCK COUNTER6/Q2: HP;  
PLACE BLOCK COUNTER6/Q0: IP;  
PLACE BLOCK COUNTER6/Q12: CP;  
PLACE BLOCK COUNTER6/\$1I160: TBUF.DQ.1;  
PLACE BLOCK COUNTER6/\$1I156: TBUF.CQ.2;  
PLACE BLOCK COUNTER6/\$1I152: TBUF.EQ.1;  
PLACE BLOCK COUNTER6/\$1I148: TBUF.DQ.2;  
PLACE BLOCK COUNTER6/\$1I144: TBUF.FQ.1;  
PLACE BLOCK COUNTER6/\$1I140: TBUF.EQ.2;  
PLACE BLOCK COUNTER6/\$1I136: TBUF.GQ.1;  
PLACE BLOCK COUNTER6/\$1I132: TBUF.FQ.2;  
PLACE BLOCK COUNTER6/\$1I128: TBUF.HQ.1;  
PLACE BLOCK COUNTER6/\$1I124: TBUF.GQ.2;  
PLACE BLOCK COUNTER6/\$1I120: TBUF.IQ.1;  
PLACE BLOCK COUNTER6/\$1I116: TBUF.HQ.2;  
PLACE BLOCK COUNTER6/\$1I112: TBUF.JQ.1;  
PLACE BLOCK COUNTER6/\$1I70: TBUF.IQ.2;  
PLACE BLOCK COUNTER5/T12: DM;  
PLACE BLOCK COUNTER5/T11: EM;  
PLACE BLOCK COUNTER5/T8: FM;  
PLACE BLOCK COUNTER5/T7: GM;  
PLACE BLOCK COUNTER5/T4: HM;  
PLACE BLOCK COUNTER5/Q10: DN;  
PLACE BLOCK COUNTER5/Q8: EN;  
PLACE BLOCK COUNTER5/Q6: FN;  
PLACE BLOCK COUNTER5/Q4: GN;  
PLACE BLOCK COUNTER5/Q2: HN;  
PLACE BLOCK COUNTER5/Q0: IN;  
PLACE BLOCK COUNTER5/Q12: CN;  
PLACE BLOCK COUNTER5/\$1I160: TBUF.DO.1;  
PLACE BLOCK COUNTER5/\$1I156: TBUF.CO.2;  
PLACE BLOCK COUNTER5/\$1I152: TBUF.EO.1;  
PLACE BLOCK COUNTER5/\$1I148: TBUF.DO.2;  
PLACE BLOCK COUNTER5/\$1I144: TBUF.FO.1;  
PLACE BLOCK COUNTER5/\$1I140: TBUF.EO.2;  
PLACE BLOCK COUNTER5/\$1I136: TBUF.GO.1;  
PLACE BLOCK COUNTER5/\$1I132: TBUF.FO.2;  
PLACE BLOCK COUNTER5/\$1I128: TBUF.HO.1;  
PLACE BLOCK COUNTER5/\$1I124: TBUF.GO.2;  
PLACE BLOCK COUNTER5/\$1I120: TBUF.IO.1;  
PLACE BLOCK COUNTER5/\$1I116: TBUF.HO.2;  
PLACE BLOCK COUNTER5/\$1I112: TBUF.JO.1;  
PLACE BLOCK COUNTER5/\$1I70: TBUF.IO.2;  
PLACE BLOCK COUNTER4/T12: DK;  
PLACE BLOCK COUNTER4/T11: EK;

PLACE BLOCK COUNTER4/T8: FK;  
PLACE BLOCK COUNTER4/T7: GK;  
PLACE BLOCK COUNTER4/T4: HK;  
PLACE BLOCK COUNTER4/Q10: DL;  
PLACE BLOCK COUNTER4/Q8: EL;  
PLACE BLOCK COUNTER4/Q6: FL;  
PLACE BLOCK COUNTER4/Q4: GL;  
PLACE BLOCK COUNTER4/Q2: HL;  
PLACE BLOCK COUNTER4/Q0: IL;  
PLACE BLOCK COUNTER4/Q12: CL;  
PLACE BLOCK COUNTER4/\$1I160: TBUF.DM.1;  
PLACE BLOCK COUNTER4/\$1I156: TBUF.CM.2;  
PLACE BLOCK COUNTER4/\$1I152: TBUF.EM.1;  
PLACE BLOCK COUNTER4/\$1I148: TBUF.DM.2;  
PLACE BLOCK COUNTER4/\$1I144: TBUF.FM.1;  
PLACE BLOCK COUNTER4/\$1I140: TBUF.EM.2;  
PLACE BLOCK COUNTER4/\$1I136: TBUF.GM.1;  
PLACE BLOCK COUNTER4/\$1I132: TBUF.FM.2;  
PLACE BLOCK COUNTER4/\$1I128: TBUF.HM.1;  
PLACE BLOCK COUNTER4/\$1I124: TBUF.GM.2;  
PLACE BLOCK COUNTER4/\$1I120: TBUF.IM.1;  
PLACE BLOCK COUNTER4/\$1I116: TBUF.HM.2;  
PLACE BLOCK COUNTER4/\$1I112: TBUF.JM.1;  
PLACE BLOCK COUNTER4/\$1I70: TBUF.IM.2;  
PLACE BLOCK COUNTER3/T12: DI;  
PLACE BLOCK COUNTER3/T11: EI;  
PLACE BLOCK COUNTER3/T8: FI;  
PLACE BLOCK COUNTER3/T7: GI;  
PLACE BLOCK COUNTER3/T4: HI;  
PLACE BLOCK COUNTER3/Q10: DJ;  
PLACE BLOCK COUNTER3/Q8: EJ;  
PLACE BLOCK COUNTER3/Q6: FJ;  
PLACE BLOCK COUNTER3/Q4: GJ;  
PLACE BLOCK COUNTER3/Q2: HJ;  
PLACE BLOCK COUNTER3/Q0: IJ;  
PLACE BLOCK COUNTER3/Q12: CJ;  
PLACE BLOCK COUNTER3/\$1I160: TBUF.DK.1;  
PLACE BLOCK COUNTER3/\$1I156: TBUF.CK.2;  
PLACE BLOCK COUNTER3/\$1I152: TBUF.EK.1;  
PLACE BLOCK COUNTER3/\$1I148: TBUF.DK.2;  
PLACE BLOCK COUNTER3/\$1I144: TBUF.FK.1;  
PLACE BLOCK COUNTER3/\$1I140: TBUF.EK.2;  
PLACE BLOCK COUNTER3/\$1I136: TBUF.GK.1;  
PLACE BLOCK COUNTER3/\$1I132: TBUF.FK.2;  
PLACE BLOCK COUNTER3/\$1I128: TBUF.HK.1;  
PLACE BLOCK COUNTER3/\$1I124: TBUF.GK.2;  
PLACE BLOCK COUNTER3/\$1I120: TBUF.IK.1;  
PLACE BLOCK COUNTER3/\$1I116: TBUF.HK.2;  
PLACE BLOCK COUNTER3/\$1I112: TBUF.JK.1;  
PLACE BLOCK COUNTER3/\$1I70: TBUF.IK.2;  
PLACE BLOCK COUNTER2/T12: DG;  
PLACE BLOCK COUNTER2/T11: EG;  
PLACE BLOCK COUNTER2/T8: FG;  
PLACE BLOCK COUNTER2/T7: GG;  
PLACE BLOCK COUNTER2/T4: HG;

PLACE BLOCK COUNTER2/Q10: DH;  
PLACE BLOCK COUNTER2/Q8: EH;  
PLACE BLOCK COUNTER2/Q6: FH;  
PLACE BLOCK COUNTER2/Q4: GH;  
PLACE BLOCK COUNTER2/Q2: HH;  
PLACE BLOCK COUNTER2/Q0: IH;  
PLACE BLOCK COUNTER2/Q12: CH;  
PLACE BLOCK COUNTER2/\$1I160: TBUF.DI.1;  
PLACE BLOCK COUNTER2/\$1I156: TBUF.CI.2;  
PLACE BLOCK COUNTER2/\$1I152: TBUF.EI.1;  
PLACE BLOCK COUNTER2/\$1I148: TBUF.DI.2;  
PLACE BLOCK COUNTER2/\$1I144: TBUF.FI.1;  
PLACE BLOCK COUNTER2/\$1I140: TBUF.EI.2;  
PLACE BLOCK COUNTER2/\$1I136: TBUF.GI.1;  
PLACE BLOCK COUNTER2/\$1I132: TBUF.FI.2;  
PLACE BLOCK COUNTER2/\$1I128: TBUF.HI.1;  
PLACE BLOCK COUNTER2/\$1I124: TBUF.GI.2;  
PLACE BLOCK COUNTER2/\$1I120: TBUF.II.1;  
PLACE BLOCK COUNTER2/\$1I116: TBUF.HI.2;  
PLACE BLOCK COUNTER2/\$1I112: TBUF.JI.1;  
PLACE BLOCK COUNTER2/\$1I70: TBUF.II.2;  
PLACE BLOCK COUNTER1/T12: DE;  
PLACE BLOCK COUNTER1/T11: EE;  
PLACE BLOCK COUNTER1/T8: FE;  
PLACE BLOCK COUNTER1/T7: GE;  
PLACE BLOCK COUNTER1/T4: HE;  
PLACE BLOCK COUNTER1/Q10: DF;  
PLACE BLOCK COUNTER1/Q8: EF;  
PLACE BLOCK COUNTER1/Q6: FF;  
PLACE BLOCK COUNTER1/Q4: GF;  
PLACE BLOCK COUNTER1/Q2: HF;  
PLACE BLOCK COUNTER1/Q0: IF;  
PLACE BLOCK COUNTER1/Q12: CF;  
PLACE BLOCK COUNTER1/\$1I160: TBUF.DG.1;  
PLACE BLOCK COUNTER1/\$1I156: TBUF.CG.2;  
PLACE BLOCK COUNTER1/\$1I152: TBUF.EG.1;  
PLACE BLOCK COUNTER1/\$1I148: TBUF.DG.2;  
PLACE BLOCK COUNTER1/\$1I144: TBUF.FG.1;  
PLACE BLOCK COUNTER1/\$1I140: TBUF.EG.2;  
PLACE BLOCK COUNTER1/\$1I136: TBUF.GG.1;  
PLACE BLOCK COUNTER1/\$1I132: TBUF.FG.2;  
PLACE BLOCK COUNTER1/\$1I128: TBUF.HG.1;  
PLACE BLOCK COUNTER1/\$1I124: TBUF.GG.2;  
PLACE BLOCK COUNTER1/\$1I120: TBUF.IG.1;  
PLACE BLOCK COUNTER1/\$1I116: TBUF.HG.2;  
PLACE BLOCK COUNTER1/\$1I112: TBUF.JG.1;  
PLACE BLOCK COUNTER1/\$1I70: TBUF.IG.2;  
PLACE BLOCK COUNTER0/T12: DC;  
PLACE BLOCK COUNTER0/T11: EC;  
PLACE BLOCK COUNTER0/T8: FC;  
PLACE BLOCK COUNTER0/T7: GC;  
PLACE BLOCK COUNTER0/T4: HC;  
PLACE BLOCK COUNTER0/Q10: DD;  
PLACE BLOCK COUNTER0/Q8: ED;  
PLACE BLOCK COUNTER0/Q6: FD;

PLACE BLOCK COUNTER0/Q4: GD;  
PLACE BLOCK COUNTER0/Q2: HD;  
PLACE BLOCK COUNTER0/Q0: ID;  
PLACE BLOCK COUNTER0/Q12: CD;  
PLACE BLOCK COUNTER0/\$1I160: TBUF.DE.1;  
PLACE BLOCK COUNTER0/\$1I156: TBUF.CE.2;  
PLACE BLOCK COUNTER0/\$1I152: TBUF.EE.1;  
PLACE BLOCK COUNTER0/\$1I148: TBUF.DE.2;  
PLACE BLOCK COUNTER0/\$1I144: TBUF.FE.1;  
PLACE BLOCK COUNTER0/\$1I140: TBUF.EE.2;  
PLACE BLOCK COUNTER0/\$1I136: TBUF.GE.1;  
PLACE BLOCK COUNTER0/\$1I132: TBUF.FE.2;  
PLACE BLOCK COUNTER0/\$1I128: TBUF.HE.1;  
PLACE BLOCK COUNTER0/\$1I124: TBUF.GE.2;  
PLACE BLOCK COUNTER0/\$1I120: TBUF.IE.1;  
PLACE BLOCK COUNTER0/\$1I116: TBUF.HE.2;  
PLACE BLOCK COUNTER0/\$1I112: TBUF.JE.1;  
PLACE BLOCK COUNTER0/\$1I70: TBUF.IE.2;  
PLACE BLOCK COUNTER15/T12: LQ;  
PLACE BLOCK COUNTER15/T11: MQ;  
PLACE BLOCK COUNTER15/T8: NQ;  
PLACE BLOCK COUNTER15/T7: OQ;  
PLACE BLOCK COUNTER15/T4: PQ;  
PLACE BLOCK COUNTER15/Q10: LR;  
PLACE BLOCK COUNTER15/Q8: MR;  
PLACE BLOCK COUNTER15/Q6: NR;  
PLACE BLOCK COUNTER15/Q4: OR;  
PLACE BLOCK COUNTER15/Q2: PR;  
PLACE BLOCK COUNTER15/Q0: QR;  
PLACE BLOCK COUNTER15/Q12: KR;  
PLACE BLOCK COUNTER15/\$1I160: TBUF.LS.1;  
PLACE BLOCK COUNTER15/\$1I156: TBUF.KS.2;  
PLACE BLOCK COUNTER15/\$1I152: TBUF.MS.1;  
PLACE BLOCK COUNTER15/\$1I148: TBUF.LS.2;  
PLACE BLOCK COUNTER15/\$1I144: TBUF.NS.1;  
PLACE BLOCK COUNTER15/\$1I140: TBUF.MS.2;  
PLACE BLOCK COUNTER15/\$1I136: TBUF.OS.1;  
PLACE BLOCK COUNTER15/\$1I132: TBUF.NS.2;  
PLACE BLOCK COUNTER15/\$1I128: TBUF.PS.1;  
PLACE BLOCK COUNTER15/\$1I124: TBUF.OS.2;  
PLACE BLOCK COUNTER15/\$1I120: TBUF.QS.1;  
PLACE BLOCK COUNTER15/\$1I116: TBUF.PS.2;  
PLACE BLOCK COUNTER15/\$1I112: TBUF.RS.1;  
PLACE BLOCK COUNTER15/\$1I70: TBUF.QS.2;  
PLACE BLOCK COUNTER14/T12: LO;  
PLACE BLOCK COUNTER14/T11: MO;  
PLACE BLOCK COUNTER14/T8: NO;  
PLACE BLOCK COUNTER14/T7: OO;  
PLACE BLOCK COUNTER14/T4: PO;  
PLACE BLOCK COUNTER14/Q10: LP;  
PLACE BLOCK COUNTER14/Q8: MP;  
PLACE BLOCK COUNTER14/Q6: NP;  
PLACE BLOCK COUNTER14/Q4: OP;  
PLACE BLOCK COUNTER14/Q2: PP;  
PLACE BLOCK COUNTER14/Q0: QP;

PLACE BLOCK COUNTER14/Q12: KP;  
PLACE BLOCK COUNTER14/\$1I160: TBUF.LQ.1;  
PLACE BLOCK COUNTER14/\$1I156: TBUF.KQ.2;  
PLACE BLOCK COUNTER14/\$1I152: TBUF.MQ.1;  
PLACE BLOCK COUNTER14/\$1I148: TBUF.LQ.2;  
PLACE BLOCK COUNTER14/\$1I144: TBUF.NQ.1;  
PLACE BLOCK COUNTER14/\$1I140: TBUF.MQ.2;  
PLACE BLOCK COUNTER14/\$1I136: TBUF.OQ.1;  
PLACE BLOCK COUNTER14/\$1I132: TBUF.NQ.2;  
PLACE BLOCK COUNTER14/\$1I128: TBUF.PQ.1;  
PLACE BLOCK COUNTER14/\$1I124: TBUF.OQ.2;  
PLACE BLOCK COUNTER14/\$1I120: TBUF.QQ.1;  
PLACE BLOCK COUNTER14/\$1I116: TBUF.PQ.2;  
PLACE BLOCK COUNTER14/\$1I112: TBUF.RQ.1;  
PLACE BLOCK COUNTER14/\$1I70: TBUF.QQ.2;  
PLACE BLOCK COUNTER13/T12: LM;  
PLACE BLOCK COUNTER13/T11: MM;  
PLACE BLOCK COUNTER13/T8: NM;  
PLACE BLOCK COUNTER13/T7: OM;  
PLACE BLOCK COUNTER13/T4: PM;  
PLACE BLOCK COUNTER13/Q10: LN;  
PLACE BLOCK COUNTER13/Q8: MN;  
PLACE BLOCK COUNTER13/Q6: NN;  
PLACE BLOCK COUNTER13/Q4: ON;  
PLACE BLOCK COUNTER13/Q2: PN;  
PLACE BLOCK COUNTER13/Q0: QN;  
PLACE BLOCK COUNTER13/Q12: KN;  
PLACE BLOCK COUNTER13/\$1I160: TBUF.LO.1;  
PLACE BLOCK COUNTER13/\$1I156: TBUF.KO.2;  
PLACE BLOCK COUNTER13/\$1I152: TBUF.MO.1;  
PLACE BLOCK COUNTER13/\$1I148: TBUF.LO.2;  
PLACE BLOCK COUNTER13/\$1I144: TBUF.NO.1;  
PLACE BLOCK COUNTER13/\$1I140: TBUF.MO.2;  
PLACE BLOCK COUNTER13/\$1I136: TBUF.OO.1;  
PLACE BLOCK COUNTER13/\$1I132: TBUF.NO.2;  
PLACE BLOCK COUNTER13/\$1I128: TBUF.PO.1;  
PLACE BLOCK COUNTER13/\$1I124: TBUF.OO.2;  
PLACE BLOCK COUNTER13/\$1I120: TBUF.QO.1;  
PLACE BLOCK COUNTER13/\$1I116: TBUF.PO.2;  
PLACE BLOCK COUNTER13/\$1I112: TBUF.RO.1;  
PLACE BLOCK COUNTER13/\$1I70: TBUF.QO.2;  
PLACE BLOCK COUNTER12/T12: LK;  
PLACE BLOCK COUNTER12/T11: MK;  
PLACE BLOCK COUNTER12/T8: NK;  
PLACE BLOCK COUNTER12/T7: OK;  
PLACE BLOCK COUNTER12/T4: PK;  
PLACE BLOCK COUNTER12/Q10: LL;  
PLACE BLOCK COUNTER12/Q8: ML;  
PLACE BLOCK COUNTER12/Q6: NL;  
PLACE BLOCK COUNTER12/Q4: OL;  
PLACE BLOCK COUNTER12/Q2: PL;  
PLACE BLOCK COUNTER12/Q0: QL;  
PLACE BLOCK COUNTER12/Q12: KL;  
PLACE BLOCK COUNTER12/\$1I160: TBUF.LM.1;  
PLACE BLOCK COUNTER12/\$1I156: TBUF.KM.2;

PLACE BLOCK COUNTER12/\$1I152: TBUF.MM.1;  
PLACE BLOCK COUNTER12/\$1I148: TBUF.LM.2;  
PLACE BLOCK COUNTER12/\$1I144: TBUF.NM.1;  
PLACE BLOCK COUNTER12/\$1I140: TBUF.MM.2;  
PLACE BLOCK COUNTER12/\$1I136: TBUF.OM.1;  
PLACE BLOCK COUNTER12/\$1I132: TBUF.NM.2;  
PLACE BLOCK COUNTER12/\$1I128: TBUF.PM.1;  
PLACE BLOCK COUNTER12/\$1I124: TBUF.OM.2;  
PLACE BLOCK COUNTER12/\$1I120: TBUF.QM.1;  
PLACE BLOCK COUNTER12/\$1I116: TBUF.PM.2;  
PLACE BLOCK COUNTER12/\$1I112: TBUF.RM.1;  
PLACE BLOCK COUNTER12/\$1I70: TBUF.QM.2;  
PLACE BLOCK COUNTER11/T12: LI;  
PLACE BLOCK COUNTER11/T11: MI;  
PLACE BLOCK COUNTER11/T8: NI;  
PLACE BLOCK COUNTER11/T7: OI;  
PLACE BLOCK COUNTER11/T4: PI;  
PLACE BLOCK COUNTER11/Q10: LJ;  
PLACE BLOCK COUNTER11/Q8: MJ;  
PLACE BLOCK COUNTER11/Q6: NJ;  
PLACE BLOCK COUNTER11/Q4: OJ;  
PLACE BLOCK COUNTER11/Q2: PJ;  
PLACE BLOCK COUNTER11/Q0: QJ;  
PLACE BLOCK COUNTER11/Q12: KJ;  
PLACE BLOCK COUNTER11/\$1I160: TBUF.LK.1;  
PLACE BLOCK COUNTER11/\$1I156: TBUF.KK.2;  
PLACE BLOCK COUNTER11/\$1I152: TBUF.MK.1;  
PLACE BLOCK COUNTER11/\$1I148: TBUF.LK.2;  
PLACE BLOCK COUNTER11/\$1I144: TBUF.NK.1;  
PLACE BLOCK COUNTER11/\$1I140: TBUF.MK.2;  
PLACE BLOCK COUNTER11/\$1I136: TBUF.OK.1;  
PLACE BLOCK COUNTER11/\$1I132: TBUF.NK.2;  
PLACE BLOCK COUNTER11/\$1I128: TBUF.PK.1;  
PLACE BLOCK COUNTER11/\$1I124: TBUF.OK.2;  
PLACE BLOCK COUNTER11/\$1I120: TBUF.QK.1;  
PLACE BLOCK COUNTER11/\$1I116: TBUF.PK.2;  
PLACE BLOCK COUNTER11/\$1I112: TBUF.RK.1;  
PLACE BLOCK COUNTER11/\$1I70: TBUF.QK.2;  
PLACE BLOCK COUNTER10/T12: LG;  
PLACE BLOCK COUNTER10/T11: MG;  
PLACE BLOCK COUNTER10/T8: NG;  
PLACE BLOCK COUNTER10/T7: OG;  
PLACE BLOCK COUNTER10/T4: PG;  
PLACE BLOCK COUNTER10/Q10: LH;  
PLACE BLOCK COUNTER10/Q8: MH;  
PLACE BLOCK COUNTER10/Q6: NH;  
PLACE BLOCK COUNTER10/Q4: OH;  
PLACE BLOCK COUNTER10/Q2: PH;  
PLACE BLOCK COUNTER10/Q0: QH;  
PLACE BLOCK COUNTER10/Q12: KH;  
PLACE BLOCK COUNTER10/\$1I160: TBUF.LI.1;  
PLACE BLOCK COUNTER10/\$1I156: TBUF.KI.2;  
PLACE BLOCK COUNTER10/\$1I152: TBUF.MI.1;  
PLACE BLOCK COUNTER10/\$1I148: TBUF.LI.2;  
PLACE BLOCK COUNTER10/\$1I144: TBUF.NI.1;

PLACE BLOCK COUNTER10/\$1I140: TBUF.MI.2;  
PLACE BLOCK COUNTER10/\$1I136: TBUF.OI.1;  
PLACE BLOCK COUNTER10/\$1I132: TBUF.NI.2;  
PLACE BLOCK COUNTER10/\$1I128: TBUF.PI.1;  
PLACE BLOCK COUNTER10/\$1I124: TBUF.OI.2;  
PLACE BLOCK COUNTER10/\$1I120: TBUF.QI.1;  
PLACE BLOCK COUNTER10/\$1I116: TBUF.PI.2;  
PLACE BLOCK COUNTER10/\$1I112: TBUF.RI.1;  
PLACE BLOCK COUNTER10/\$1I70: TBUF.QI.2;  
PLACE BLOCK COUNTER9/T12: LE;  
PLACE BLOCK COUNTER9/T11: ME;  
PLACE BLOCK COUNTER9/T8: NE;  
PLACE BLOCK COUNTER9/T7: OE;  
PLACE BLOCK COUNTER9/T4: PE;  
PLACE BLOCK COUNTER9/Q10: LF;  
PLACE BLOCK COUNTER9/Q8: MF;  
PLACE BLOCK COUNTER9/Q6: NF;  
PLACE BLOCK COUNTER9/Q4: OF;  
PLACE BLOCK COUNTER9/Q2: PF;  
PLACE BLOCK COUNTER9/Q0: QF;  
PLACE BLOCK COUNTER9/Q12: KF;  
PLACE BLOCK COUNTER9/\$1I160: TBUF.LG.1;  
PLACE BLOCK COUNTER9/\$1I156: TBUF.KG.2;  
PLACE BLOCK COUNTER9/\$1I152: TBUF.MG.1;  
PLACE BLOCK COUNTER9/\$1I148: TBUF.LG.2;  
PLACE BLOCK COUNTER9/\$1I144: TBUF.NG.1;  
PLACE BLOCK COUNTER9/\$1I140: TBUF.MG.2;  
PLACE BLOCK COUNTER9/\$1I136: TBUF.OG.1;  
PLACE BLOCK COUNTER9/\$1I132: TBUF.NG.2;  
PLACE BLOCK COUNTER9/\$1I128: TBUF.PG.1;  
PLACE BLOCK COUNTER9/\$1I124: TBUF.OG.2;  
PLACE BLOCK COUNTER9/\$1I120: TBUF.QG.1;  
PLACE BLOCK COUNTER9/\$1I116: TBUF.PG.2;  
PLACE BLOCK COUNTER9/\$1I112: TBUF.RG.1;  
PLACE BLOCK COUNTER9/\$1I70: TBUF.QG.2;  
PLACE BLOCK COUNTER8/T12: LC;  
PLACE BLOCK COUNTER8/T11: MC;  
PLACE BLOCK COUNTER8/T8: NC;  
PLACE BLOCK COUNTER8/T7: OC;  
PLACE BLOCK COUNTER8/T4: PC;  
PLACE BLOCK COUNTER8/Q10: LD;  
PLACE BLOCK COUNTER8/Q8: MD;  
PLACE BLOCK COUNTER8/Q6: ND;  
PLACE BLOCK COUNTER8/Q4: OD;  
PLACE BLOCK COUNTER8/Q2: PD;  
PLACE BLOCK COUNTER8/Q0: QD;  
PLACE BLOCK COUNTER8/Q12: KD;  
PLACE BLOCK COUNTER8/\$1I160: TBUF.LE.1;  
PLACE BLOCK COUNTER8/\$1I156: TBUF.KE.2;  
PLACE BLOCK COUNTER8/\$1I152: TBUF.ME.1;  
PLACE BLOCK COUNTER8/\$1I148: TBUF.LE.2;  
PLACE BLOCK COUNTER8/\$1I144: TBUF.NE.1;  
PLACE BLOCK COUNTER8/\$1I140: TBUF.ME.2;  
PLACE BLOCK COUNTER8/\$1I136: TBUF.OE.1;  
PLACE BLOCK COUNTER8/\$1I132: TBUF.NE.2;

```
PLACE BLOCK COUNTER8/$1I128: TBUF.PE.1;
PLACE BLOCK COUNTER8/$1I124: TBUF.OE.2;
PLACE BLOCK COUNTER8/$1I120: TBUF.QE.1;
PLACE BLOCK COUNTER8/$1I116: TBUF.PE.2;
PLACE BLOCK COUNTER8/$1I112: TBUF.RE.1;
PLACE BLOCK COUNTER8/$1I70: TBUF.QE.2;
PLACE BLOCK -WRT_EF4: M14;
PLACE BLOCK -WRT_EF3: R16;
PLACE BLOCK -WRT_EF2: N15;
PLACE BLOCK -WRT_EF1: P16;
PLACE BLOCK -WRT_EF0: N16;
PLACE BLOCK -RD_EF4: L14;
PLACE BLOCK -RD_EF3: M15;
PLACE BLOCK -RD_EF2: M16;
PLACE BLOCK -RD_EF1: L15;
PLACE BLOCK -RD_EF0: L16;
# End
```

### ***10.3 Placement Report***

**Error! Not a valid filename.**

#### ***10.4 XACT Performance***

## 10.5 State CAD Diagrams

### 10.5.1 VME

\cdf\src\err\_log\vme.dia  
ERR\_LOG VME decoder  
10/1/96 JO  
Revised 8/1/97 NF

```
ADDR[] = 5:0

CREG = REQ & ADDR[0]
STS_REG = REQ & !WRITE & ADDR[^H2]
MASK = REQ & !RUN & ADDR[^H4]

FIFO0 = REQ & !RUN & !WRITE & ADDR[^H6]           FIFO_TMP = REQ & !RUN & !WRITE & ADDR[^H8]
FIFO2 = REQ & !RUN & !WRITE & ADDR[^H8]           FIFO_TMP IS TO ALLOW READING OF THE SRC AND VRB ERROR FIFOS.
                                                     THEY BOTH USE DATA LINES 0-7 NOT 0-15
FIFO4 = REQ & !RUN & !WRITE & ADDR[^HC]

CNT0 = REQ & !RUN & !WRITE & ADDR[^H10]
CNT1 = REQ & !RUN & !WRITE & ADDR[^H12]
CNT2 = REQ & !RUN & !WRITE & ADDR[^H14]
CNT3 = REQ & !RUN & !WRITE & ADDR[^H16]
CNT4 = REQ & !RUN & !WRITE & ADDR[^H18]
CNT5 = REQ & !RUN & !WRITE & ADDR[^H1A]
CNT6 = REQ & !RUN & !WRITE & ADDR[^H1C]
CNT7 = REQ & !RUN & !WRITE & ADDR[^H1E]
CNT8 = REQ & !RUN & !WRITE & ADDR[^H20]
CNT9 = REQ & !RUN & !WRITE & ADDR[^H22]
CNT10 = REQ & !RUN & !WRITE & ADDR[^H24]
CNT11 = REQ & !RUN & !WRITE & ADDR[^H26]
CNT12 = REQ & !RUN & !WRITE & ADDR[^H28]
CNT13 = REQ & !RUN & !WRITE &
CNT14 = REQ & !RUN & !WRITE &
CNT15 = REQ & !RUN & !WRITE &

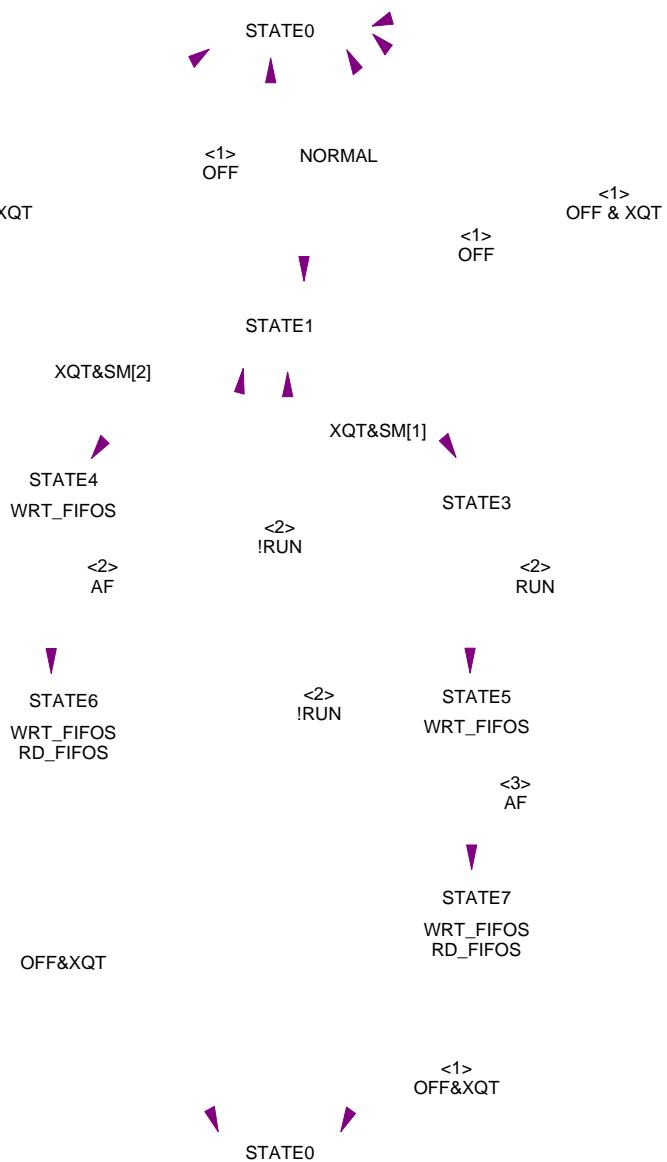
ACK = REQ & ADDR[0] #
REQ & ADDR[2] & !WRITE #
REQ & ADDR[4] & !RUN #
REQ & !WRITE & !RUN & (ADDR<^H30)
```

## 10.5.2 RUNMODES

\cdflsrc\err\_log\runmodes.dia  
 Sept. 30, 1996  
 J.Oliver  
 (View:70% portrait)

MODE[] = CREG6 CREG5 CREG4  
 SM[] = CREG3 CREG2 CREG1 CREG0  
 %OFF% = MODE[0]  
 %NORMAL% = MODE[3]

OFF&SM[1]&XQT STATE2  
 RES\_FIFOS



## 10.6 HDL Code

### 10.6.1 VME

```
" D:\CDF\SRC\ERR_LOG\VME.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.10
" Fri Aug 29 13:56:43 1997

" This Abel code was generated using:
" binary encoded state assignment with structured code format.
" Minimization is enabled, implied else is enabled,
" and outputs are speed optimized.
```

MODULE VME

DECLARATIONS

"Input variables

```
ADDR0 PIN;
ADDR1 PIN;
ADDR2 PIN;
ADDR3 PIN;
ADDR4 PIN;
ADDR5 PIN;
REQ PIN;
RUN PIN;
WRITE PIN;
```

"Logic variables

```
ACK PIN ISTYPE 'com';
CNT0 PIN ISTYPE 'com';
CNT1 PIN ISTYPE 'com';
CNT2 PIN ISTYPE 'com';
CNT3 PIN ISTYPE 'com';
CNT4 PIN ISTYPE 'com';
CNT5 PIN ISTYPE 'com';
CNT6 PIN ISTYPE 'com';
CNT7 PIN ISTYPE 'com';
CNT8 PIN ISTYPE 'com';
CNT9 PIN ISTYPE 'com';
CNT10 PIN ISTYPE 'com';
CNT11 PIN ISTYPE 'com';
CNT12 PIN ISTYPE 'com';
CNT13 PIN ISTYPE 'com';
CNT14 PIN ISTYPE 'com';
CNT15 PIN ISTYPE 'com';
CREG PIN ISTYPE 'com';
FIFO0 PIN ISTYPE 'com';
FIFO1 PIN ISTYPE 'com';
FIFO2 PIN ISTYPE 'com';
```

```

FIFO4 PIN ISTYPE 'com';
MASK PIN ISTYPE 'com';
STS_REG PIN ISTYPE 'com';

"Vectors
DECLARATIONS
ADDR=[
    ADDR5,
    ADDR4,
    ADDR3,
    ADDR2,
    ADDR1,
    ADDR0
];
"Logic Equations
EQUATIONS
ACK = !ADDR5 & !ADDR4 & !ADDR3 & !ADDR2 & !ADDR1 & !ADDR0 & REQ # !WRITE &
    !ADDR5 & !ADDR4 & !ADDR3 & !ADDR2 & !ADDR0 & REQ # !RUN & !ADDR5 &
    !ADDR4
        & !ADDR3 & !ADDR1 & !ADDR0 & REQ # !ADDR5 & REQ & !WRITE & !RUN #
    !ADDR4
        & REQ & !WRITE & !RUN ;
CNT0 = !ADDR5 & ADDR4 & !ADDR3 & !ADDR2 & !ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;
CNT1 = !ADDR5 & ADDR4 & !ADDR3 & !ADDR2 & ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;
CNT2 = !ADDR5 & ADDR4 & !ADDR3 & ADDR2 & !ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;
CNT3 = !ADDR5 & ADDR4 & !ADDR3 & ADDR2 & ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;
CNT4 = !ADDR5 & ADDR4 & ADDR3 & !ADDR2 & !ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;
CNT5 = !ADDR5 & ADDR4 & ADDR3 & !ADDR2 & ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;
CNT6 = !ADDR5 & ADDR4 & ADDR3 & ADDR2 & !ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;
CNT7 = !ADDR5 & ADDR4 & ADDR3 & ADDR2 & ADDR1 & !ADDR0 & REQ & !RUN & !WRITE
    ;
CNT8 = ADDR5 & !ADDR4 & !ADDR3 & !ADDR2 & !ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;
CNT9 = ADDR5 & !ADDR4 & !ADDR3 & !ADDR2 & ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;
CNT10 = ADDR5 & !ADDR4 & !ADDR3 & ADDR2 & !ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;
CNT11 = ADDR5 & !ADDR4 & !ADDR3 & ADDR2 & ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;
CNT12 = ADDR5 & !ADDR4 & ADDR3 & !ADDR2 & !ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;
CNT13 = ADDR5 & !ADDR4 & ADDR3 & !ADDR2 & ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;
CNT14 = ADDR5 & !ADDR4 & ADDR3 & ADDR2 & !ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;
CNT15 = ADDR5 & !ADDR4 & ADDR3 & ADDR2 & ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;

```

```
CREG = !ADDR5 & !ADDR4 & !ADDR3 & !ADDR2 & !ADDR1 & !ADDR0 & REQ ;
FIFO0 = !ADDR5 & !ADDR4 & !ADDR3 & ADDR2 & ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;
FIFO1 = !ADDR5 & !ADDR4 & ADDR3 & !ADDR2 & !ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;
FIFO2 = !ADDR5 & !ADDR4 & ADDR3 & !ADDR2 & ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;
FIFO4 = !ADDR5 & !ADDR4 & ADDR3 & ADDR2 & !ADDR1 & !ADDR0 & REQ & !RUN & !
    WRITE ;
MASK = !ADDR5 & !ADDR4 & !ADDR3 & ADDR2 & !ADDR1 & !ADDR0 & REQ & !RUN ;
STS_REG = !ADDR5 & !ADDR4 & !ADDR3 & !ADDR2 & ADDR1 & !ADDR0 & REQ & !WRITE
;
END VME
```

## 10.6.2 RUNMODES

```
" D:\CDF\SRC\ERR_LOG\RUNMODES.ABL
" ABEL code created by Visual Software Solution's StateCAD Version 3.02c0
" Tue Oct 01 09:23:47 1996

" This Abel code was generated using:
" binary encoded state assignment with boolean code format.
" Minimization is enabled, implied else is enabled,
" and outputs are speed optimized.
```

### MODULE RUNMODES

#### DECLARATIONS

```
"clock name
    CLK PIN;
```

#### "Input variables

```
    AF PIN;
    CREG0 PIN;
    CREG1 PIN;
    CREG2 PIN;
    CREG3 PIN;
    CREG4 PIN;
    CREG5 PIN;
    CREG6 PIN;
    RUN PIN;
    XQT PIN;
```

#### "Output variables

```
    RD_FIFOS PIN ISTYPE 'reg';
    RES_FIFOS PIN ISTYPE 'reg';
    WRT_FIFOS PIN ISTYPE 'reg';
```

#### "State variables

```
    SV0 PIN ISTYPE 'reg';
    SV1 PIN ISTYPE 'reg';
    SV2 PIN ISTYPE 'reg';
```

#### "Vectors

#### DECLARATIONS

```
    MODE=[
        CREG6,
        CREG5,
        CREG4
    ];
```

```
    SM=[
        CREG3,
        CREG2,
        CREG1,
```

```

        CREG0
    };

"Clocked logic clock setup
EQUATIONS
    RD_FIFOS.clk=CLK;
    RES_FIFOS.clk=CLK;
    WRT_FIFOS.clk=CLK;

"State Register assignment
DECLARATIONS
    sreg=[ SV0,SV1,SV2];

EQUATIONS
    sreg.clk=CLK;

DECLARATIONS
    STATE0=[0, 0, 0];
    STATE1=[0, 0, 1];
    STATE2=[0, 1, 0];
    STATE3=[0, 1, 1];
    STATE4=[1, 0, 0];
    STATE5=[1, 0, 1];
    STATE6=[1, 1, 0];
    STATE7=[1, 1, 1];

EQUATIONS
SV0 := !CREG3 & !CREG2 & CREG1 & !CREG0 & XQT & !SV0.FB & !SV1.FB & SV2.FB # 
    !XQT & SV0.FB & !SV2.FB # CREG6 & SV0.FB & !SV2.FB # CREG5 & SV0.FB & !
    SV2.FB # CREG4 & SV0.FB & !SV2.FB # CREG4 & RUN & SV0.FB & !SV1.FB #
    CREG5 & RUN & SV0.FB & !SV1.FB # CREG6 & RUN & SV0.FB & !SV1.FB # RUN &
    CREG4 & SV1.FB & SV2.FB # RUN & CREG5 & SV1.FB & SV2.FB # RUN & CREG6 &
    SV1.FB & SV2.FB # RUN & !XQT & SV0.FB ;

SV1 := XQT & !CREG3 & !CREG2 & !CREG1 & CREG0 & !CREG6 & !CREG5 & !CREG4 & !
    SV0.FB & !SV1.FB # !CREG3 & !CREG2 & !CREG1 & CREG0 & XQT & !SV0.FB & !
    SV1.FB & SV2.FB # !RUN & CREG4 & !SV0.FB & SV1.FB & SV2.FB # !RUN & CREG5
    & !SV0.FB & SV1.FB & SV2.FB # !RUN & CREG6 & !SV0.FB & SV1.FB & SV2.FB #
    CREG4 & AF & SV0.FB & !SV2.FB # CREG5 & AF & SV0.FB & !SV2.FB # CREG6 &
    AF & SV0.FB & !SV2.FB # !XQT & AF & SV0.FB & !SV2.FB # !XQT & SV0.FB &
    SV1.FB & !SV2.FB # CREG6 & SV0.FB & SV1.FB & !SV2.FB # CREG5 & SV0.FB &
    SV1.FB & !SV2.FB # CREG4 & SV0.FB & SV1.FB & !SV2.FB # CREG4 & RUN & AF &
    SV0.FB # CREG5 & RUN & AF & SV0.FB # CREG6 & RUN & AF & SV0.FB # !XQT
    & RUN & AF & SV0.FB # RUN & CREG4 & SV0.FB & SV1.FB # RUN & CREG5 &
    SV0.FB & SV1.FB # RUN & CREG6 & SV0.FB & SV1.FB # RUN & !XQT & SV0.FB &
    SV1.FB ;

SV2 := !CREG6 & CREG5 & CREG4 & !SV0.FB & !SV1.FB & !SV2.FB # CREG3 & CREG4
    & SV2.FB # !CREG0 & !CREG1 & CREG4 & SV2.FB # CREG1 & CREG0 & CREG4 &
    SV2.FB # !XQT & CREG4 & SV2.FB # CREG3 & CREG5 & SV2.FB # !CREG0 & !
    CREG1 & CREG5 & SV2.FB # CREG1 & CREG0 & CREG5 & SV2.FB # !XQT & CREG5 &
    SV2.FB # CREG3 & CREG6 & SV2.FB # !CREG0 & !CREG1 & CREG6 & SV2.FB #
    CREG1 & CREG0 & CREG6 & SV2.FB # !XQT & CREG6 & SV2.FB # CREG2 & CREG4 &
```

```

SV2.FB # CREG2 & CREG5 & SV2.FB # CREG2 & CREG6 & SV2.FB # !CREG3 & !
CREG2 & !CREG1 & CREG0 & XQT & !SV0.FB & !SV1.FB & SV2.FB # CREG4 & SV0.FB
& SV2.FB # CREG5 & SV0.FB & SV2.FB # CREG6 & SV0.FB & SV2.FB # CREG4 &
SV1.FB & SV2.FB # CREG5 & SV1.FB & SV2.FB # CREG6 & SV1.FB & SV2.FB # !
XQT & SV0.FB & SV2.FB ;

RD_FIFOS := CREG4 & AF & SV0.FB & !SV2.FB # CREG5 & AF & SV0.FB & !SV2.FB
# CREG6 & AF & SV0.FB & !SV2.FB # !XQT & AF & SV0.FB & !SV2.FB # !XQT &
SV0.FB & SV1.FB & !SV2.FB # CREG6 & SV0.FB & SV1.FB & !SV2.FB # CREG5 &
SV0.FB & SV1.FB & !SV2.FB # CREG4 & SV0.FB & SV1.FB & !SV2.FB # CREG4 &
RUN & AF & SV0.FB # CREG5 & RUN & AF & SV0.FB # CREG6 & RUN & AF & SV0.FB
# !XQT & RUN & AF & SV0.FB # RUN & CREG4 & SV0.FB & SV1.FB # RUN &
CREG5 & SV0.FB & SV1.FB # RUN & CREG6 & SV0.FB & SV1.FB # RUN & !XQT &
SV0.FB & SV1.FB ;

RES_FIFOS := XQT & !CREG3 & !CREG2 & !CREG1 & CREG0 & !CREG6 & !CREG5 & !
CREG4 & !SV0.FB & !SV1.FB & !SV2.FB ;

WRT_FIFOS := !CREG3 & !CREG2 & CREG1 & !CREG0 & XQT & !SV0.FB & !SV1.FB &
SV2.FB # !XQT & SV0.FB & !SV2.FB # CREG6 & SV0.FB & !SV2.FB # CREG5 &
SV0.FB & !SV2.FB # CREG4 & SV0.FB & !SV2.FB # CREG4 & RUN & SV0.FB & !
SV1.FB # CREG5 & RUN & SV0.FB & !SV1.FB # CREG6 & RUN & SV0.FB & !SV1.FB
# RUN & CREG4 & SV1.FB & SV2.FB # RUN & CREG5 & SV1.FB & SV2.FB # RUN &
CREG6 & SV1.FB & SV2.FB # RUN & !XQT & SV0.FB ;
END RUNMODES

```

## **11. Schematics**

